Dialect Classification and Multi-Dialect Speech Recognition

Thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy
in
Computer Science and Engineering

by

Rashmi Kethireddy
20172044
rashmi.kethireddy@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500032, India
February, 2024

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Dialect Classification and Multi-Dialect Speech Recognition**" by **Rashmi Kethireddy** (Roll No. 20172044), has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Dr. Suryakanth V Gangashetty

Acknowledgments

First, I would like to thank and express my profound respect to my guide Dr Suryakanth-sir. His friendly nature, unconventional thinking about research, unconditional support, and enthusiasm allowed me to pursue research with complete freedom. I also thank him for his financial support in research.

I am highly indebted to Dr Sudarsana Reddy Kadiri (sir) for being a mentor and well-wisher. I sincerely thank for his unconditional support and constant mentoring throughout my research. Without his encouragement and guidance, I wouldn't have reached this stage of submitting my PhD thesis. I also thank Prof. Paavo Alku for the collaboration and transfer of knowledge.

I express my deepest respect and sincere gratitude to Prof. B. Yegnanarayana for his lectures on speech signal processing. I also thank Prof. Preethi Jyothi for her lectures on speech recognition.

I thank my comprehensive reviewers, Dr Anil sir and Prof. B. Yegnanarayana. I would also thank my reviewer of PhD thesis proposal, Dr Anil sir and Prof. Priyankoo Sarmah. Their critical inputs have helped in developing the thesis.

I want to thank my seniors in PhD, Bhanu sir, Santosh Kesiraju sir, Gangamohan sir, Vishala, and Ravishankar Prasad sir, who was always a call away for guidance.

I want to thank my hangout buddies, Tirusha, Murtuza, Nikitha, Suman, and Nitin sir. I am left with excellent memories of IIITH. Not only a friendly company but also many fruitful discussions that increased my research interest.

I would like to thank Kavitha, ma'am, for the lectures on cognitive neuroscience. Being one of the active persons on campus, she is an inspiration to me. I would like to thank Prof. Krishna Reddy for the lectures on database systems. I would also like to thank Dr Vineeth Gandhi and Dr Sujit Gujar for their incredible lectures, which gave more significant insights into statistical methods in AI and optimization methods.

I cherish a special bond with Sushmita ma'am, Vijayalakshmi ma'am, Haala, Rambabu sir, Kishore Botsa, Rashmi Kalshetty, Krishna, Ganesh, Poorva, Chandrashekar sir, Nayan Anand and Jhansi.

Special thanks to UGC-NET, who financed my research. I also thank the staff of IIITH, especially, Pushpalatha ma'am, Prathima ma'am, Rambabu sir, and Shakmukh sir, for their quick response.

I want to thank all my teachers and friends who have helped me reach this stage in this life. I especially thank my guide in M.Tech. Dr Suresh Lokhande, sir, for his mentorship, which leads to my choosing PhD.

This endeavour would not have been possible without my husband, Pavan and son, Sricharan. I would like to thank them for patiently bearing with me during my PhD years. I also would like to thank them for their unconditional love. Words cannot express my gratitude to my parents and my in-laws. Their belief in me has kept my spirits and motivation high during this process.

I want to thank my friends, Sita, Lalitha, Deepthi, and Deepika, for the love and affection they showered on me. After their friendship, my perspective on life has changed.

-Rashmi Kethireddy

Abstract

Keywords: *dialect classification; zero-time windowing; single frequency filtering; frequency domain linear prediction; convolution neural network; ECAPA-TDNN; deepspeech; multi-dialect automatic speech recognition; Indian English ASR*

Major goal of this thesis is to study the dialectal variations and improve the performance of speech recognition with an embeddings derived from improved dialect classification system. Initial studies focused on improvement of dialect classification system with three major dialects (AU:Australian, UK:Britain, and US:American) of English.

In order to improve the performance of dialect classification system and based on the analysis of dialectal variations, advanced signal processing approaches were proposed to investigate for dialect classification with traditional i-vector system. The features that provide high spectral resolution will help to capture subtle differences between dialects. So, this thesis proposed to use single frequency filtering (SFF) and zero-time windowing (ZTW) based features that provide high spectral resolution without compromising temporal resolution. Along with frame level spectral resolution, longer temporal context will constitute for dialect classification. So, approaches that enhance the temporal context of proposed features (SFF and ZTW) approaches such as delta and double delta coefficients (Δ+$\Delta\Delta$), shifted delta coefficients (SDCs) are experimented. It is observed that dialect classification system has given promising performance with the proposed features with temporal context provided by Δ+$\Delta\Delta$ and SDCs. Further, signal processing approaches that can provide long temporal summarization such as frequency domain linear prediction (FDLP) are proposed for dialect classification. From experiments, with FDLP based features, it is observed that long temporal summarization provided by FDLP based features is advantageous for discriminating dialects. So, both the signal processing approaches that provide high spectral resolution (SFF and ZTW) and long temporal summarization (FDLP) have shown to give promising performance in dialect classification when compared to commonly used STFT based features.

Further, due to promising performance by deep neural networks in classification tasks and its ability to provide longer temporal context, simpler (CNN) to advanced deep neural network (TCN, TDNN, and ECAPA-TDNN) architectures that provide different temporal contexts are investigated, it is observed that

advanced neural network architectures improved the performance of dialect classification. Further, on evaluation of the best of both stages, it is observed that ECAPA-TDNN performed better with proposed features (SFF).

The dialectal variations in speech degrade the performance of multi-dialectal automatic speech recognition (ASR) system. The embeddings derived from the best dialect classification system are applied to multi-dialect (with AU, UK, and US dialects) ASR and found to improve the performance of the ASR system.

In most studies, Indian English is considered as a single dialect even though it has different native speakers. So, the inclusion of foreign dialectal embeddings improved the performance of the ASR system. The observations made in dialect classification systems with major dialects of English are extended to foreign dialect classification (i.e., native language (or L1) identification). The embeddings extracted from the improved dialect classification system are included along with the Indian English ASR system to improve the performance.

Contents

List of Figures

List of Tables

Abbreviations

Δ Coefficients	- Delta Coefficients
ΔΔ Coefficients	- Double Delta Coefficients
ASR	- Automatic Speech Recognition
CNN	- Convolution Neural Network
Conv1D	- One-dimensional convolution neural network
CTC	- Connectionist Temporal Classification
DCT	- Discrete Cosine Transform
DNN	- Deep Neural Network
ECAPA-TDNN	- Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network
EM	- Expectation Maximization
FDLP	- Frequency Domain Linear Prediction
FDLPCC	- Frequency Domain Linear Prediction Cepstral Coefficients
FFT	- Fast Fourier Transform
GMM	- Gaussian Mixture Model
GRU	- Gated Recurrent Unit
HMM	- Hidden Markov Model
JFA	- Joint Factor Analysis
KL divergence	- Kullback-Leibler divergence
L1	- Native language of speaker
L2	- Second/Acquired language of speaker
LPC	- Linear Prediction Coefficients
LPCC	- Linear Prediction Cepstral Coefficients
LSTM	- Long Short Term Memory
MAP	- Maximum Aposteriori Adaptation
MFBE	- Mel Filter Bank Energies
MFCC	- Mel Frequency Cepstral Coefficients

MFCC-SFF	- Mel Frequency Cepstral Coefficients derived from Single Frequency Filtering spectrum
MFCC-STFT	- Mel Frequency Cepstral Coefficients derived from Short Time Fourier Transform spectrum
MFCC-ZTW	- Mel Frequency Cepstral Coefficients derived from Zero Time Windowing spectrum
ML	- Maximum Likelihood
MLLR	- Maximum Likelihood Linear Regression
MSE	- Mean Square Error
OOV	- Out of Vocabulary
PCA	- Principal Component Analysis
PPRLM	- Parallel Phone Recognition followed by Language Modeling
PP	- Perplexity
ReLU	- Rectified Linear Unit
RNN	- Recurrent Neural Network
SDC	- Shifted Delta Coefficients
SFF	- Single Frequency Filtering
SFFCC	- Single Frequency Filtering Cepstral Coefficients
SGD	- Stochastic Gradient Descent
SP	- Speed Perturbation
STFT	- Short Time Fourier Transform
SVM	- Support Vector Machine
SVP	- Speed and Volume Perturbation
TCNN	- Temporal Convolution Neural Network
TDNN	- Time Delay Neural Network
t-SNE	- t-distributed Stochastic Neighbor Embedding
UAR	- Unweighted Average Recall
UBM	- Universal Background Model
VP	- Volume Perturbation
WER	- Word Error Rate
ZTW	- Zero Time Windowing
ZTWCC	- Zero Time Windowed Cepstral Coefficients

Chapter 1

Introduction

Speech signals not only convey linguistic information (the message) but also carry information related to other extrinsic and intrinsic characteristics. Extrinsic characteristics include environmental conditions during recording, such as noise and other speaker interference. Intrinsic characteristics include internal factors related to speaker variability, and it can occur in speech due to emotional state, physiology, speaking style or rate of speaking of the speaker, age, and dialect.

Speaker acquires common patterns from people around him/her that result in language dialects. Over time, he/she habituates these patterns, resulting in constrained articulatory movements that, in turn, lead to pronunciation variations. Dialects are of three types: social, regional, and foreign. The patterns in speech acquired by a speaker due to his/her social conditions are social dialects, and the patterns in a speaker's speech acquired based on geographic location are regional dialects. Speakers also exhibit linguistic constraints posed by the first language in the second language resulting in foreign dialects.

Out of all these variabilities, it is observed in [4] that gender and accent are the first two principal components of variability. So, this thesis focuses on one of the most variable components, dialect. The dialectal variations can be observed at three levels, and they are at pronunciation, vocabulary, and grammar. Pronunciation variations can be observed due to differences in phonetic realization, phonotactic distribution, phonemic system, and prosodic characteristics [5]. Phonetic realization differs between dialects, leading to phonemes' addition, deletion, and insertion.

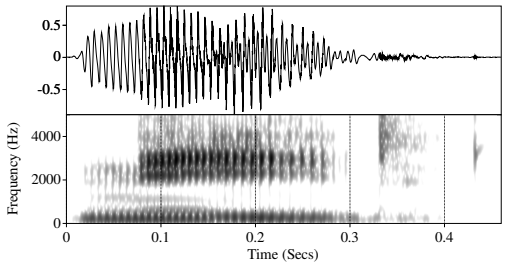
Phonotactic distribution is the pattern in the phonological structure corresponding to a language or dialect. It varies between dialects due to their rules constraining the occurrence of two phonemes together. Due to the rule defined in Southern US English, the /*i*/ and / ɛ/ are neutralized before nasals, which will result in the pronunciation of [*pin*] for both the words pin and pen. The difference in the phonemic alphabetic system between British English (uses Cambridge University's BEEP dictionary) and American English (uses CMU's dictionary) leads to confusion about phoneme pronunciation.

The phonemic system changes between regional and foreign accents because of the differences in their phonemic inventory (number of phonemes or identity of phonemes). We can illustrate this by looking at a

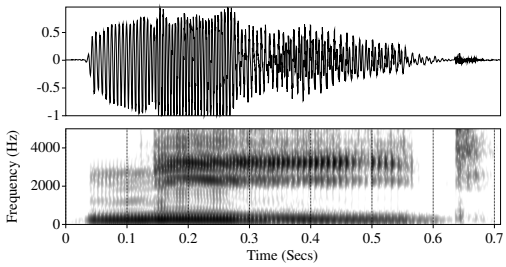
Chinese speaker whose phoneme inventory does not include the phoneme [æ]. Therefore, Chinese speakers tend to replace the phone with a similar-sounding phone.

Prosodic characteristics exist at different levels: stress, duration, and intonation [6]. For illustrations to observe prosodic variations, American (US) and British (UK) English are considered. For stress variations, the French loan word "Garage" is considered. "Adult" is one of the loan words from French. Such words were adapted differently between US and UK dialects. Americans emphasize on second syllable while the British emphasize on first syllable [7]. The stress assignment for the word "adult" is [ʌdʌ'lt] for American speaker and [ʌ'dʌlt] for British speaker.

For the durational variations, the short-time Fourier transform (STFT) spectrograms of the word "need" is plotted in Figure 1.1 for UK and US dialects. It can be observed from the spectrograms of the UK dialect (Figure 1.1 (a)) and US dialect (Figure 1.1 (a)) that the US dialect took longer duration to pronounce /i/ compared to the UK dialect.



(a) STFT Spectrogram of "need" by **UK dialect**

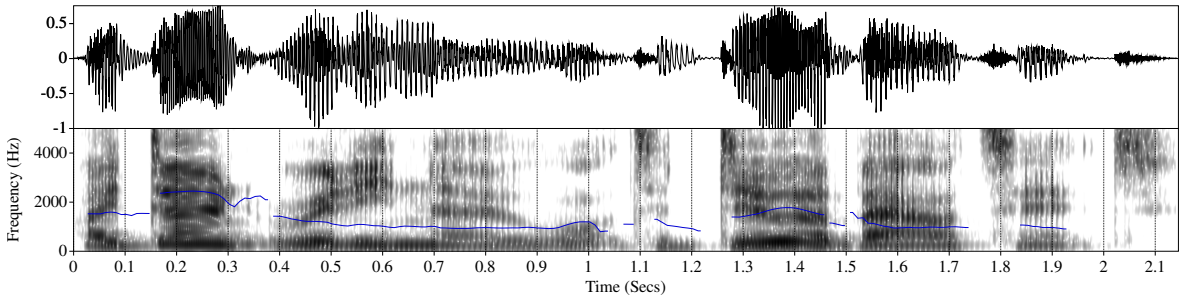


(b) STFT Spectrogram of "need" by **US dialect**

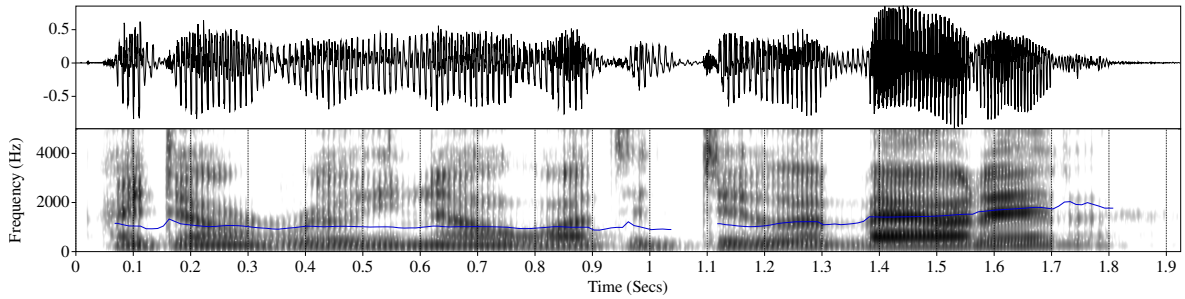
Figure 1.1: Illustration of variations in duration across UK and US dialects using STFT spectrograms for the word "need".

The intonation or melody of American and British speakers vary. Figure 1.2 shows the spectrogram for the sentence "I don't really know what to do about it". In principle, UK speakers lower the intonation after the main stress while US speakers tend to upspeak after the main stress. The fundamental frequency (F1) contour is shown as a blue line in the figures. It can be observed that the F1 drops after the main stress in the case of the UK but in the US the F1 is raised after the main stressed word.

Variations in vocabulary, spelling and grammar across dialects will result in variations in the lexical distribution of characters and words. People in Spanish use *mobile* while Latin Americans use *cellular* for the word *phone*. Spelling variations such as "colour" in British English vs "color" in American English can also be observed between dialects. One of the grammatical variations is that Newcastle has a plural form for the second person pronoun "you", "yous". Semantic variations across dialects have different mappings



(a) STFT Spectrogram of **UK dialect**



(b) STFT Spectrogram of **US dialect**

Figure 1.2: Illustration of intonational variations across UK and US dialects using STFT spectrograms for the sentence "I don't really know what to do about it.".

between a word to its meaning. These differences either map more than one meaning to a word or more than one word to meaning.

1.1 Dialectal challenges in speech recognition system

A listener not only tries to understand the underlying linguistic content but also extracts various aspects of a speaker by perceiving the speaker's speech. One such aspect of the speaker is the regional origin and native language of the speaker. Even though these factors modulate the speech signal, a listener can easily extract linguistic information. However, these variations lead to misrecognition in speech recognition systems.

Usually, language-specific speech applications are developed, and these dialectal variations within a language degrade the performance of speech applications. Dialectal variations in speech can influence the performance of automatic speech recognition (ASR) systems [8–10]. In [8], cross-dialect models were found

to increase the error rate by 40-50% (relative) compared to dialect-specific models. With the advancement of speech recognition systems in day-to-day life, avoiding these disparities has become the need of the hour. In a study [11], five popular commercial ASR systems were tested with people from different races. The performance gaps suggest it is harder for a group of people to benefit from the increasingly widespread use of speech recognition technology. So, research to improve the performance of speech systems despite social/regional/foreign dialectal variations is required.

The current speech recognition system mainly has three modules-acoustic model, pronunciation model, and language model. The differences in phonetic variations can lead to errors in the speech recognition system's acoustic or pronunciation model. In tonal languages, the meaning of word changes with pitch and these tones vary across dialects leading to misinterpretation of the word. Variations in vocabulary and grammar can lead to the absence of words in the language model, which could lead to out-of-vocabulary (OOV).

Three solutions were presented in the literature to deal with such situations in the acoustic model (maps sequence of feature representations to phonetic transcription) of an ASR system. They are, unified acoustic model [8, 9], dialect dependent acoustic model [8], and dialect adaptation methods [9, 12]. In a unified acoustic model, ASR models were trained with data from all dialects of data and learned the traits of all dialects collectively [13]. The challenging part of developing efficient multi-dialectal acoustic models is to have an enormous amount of dialect-specific data. In [8, 9], it is shown that other solutions performed better than the multi-dialectal acoustic model. In dialect-dependent acoustic models [8, 14], a separate acoustic model is trained for every class of dialect. In [8], it is shown that dialect dependent model with a good dialect identification system prior is better than dialect adaptation techniques provided with the required amount of speech corpus. In acoustic model adaptation approaches, various techniques such as maximum aposteriori adaptation/maximum likelihood linear regression (MAP/MLLR) adaptation of traditional HMM-GMM acoustic model [15, 16], fine-tuning acoustic models to specific accents [13, 14], continuous accentedness score [15], the inclusion of accent embeddings [9, 10, 12, 15, 17], and joint training of dialect and speech recognition systems [9, 12] were carried out to improve the performance. Experiments have revealed that the inclusion of dialect information either through embeddings or joint modelling of both the systems has been shown to give promising results with ASR system [9, 10, 12, 15].

Dialectal variations also influence the performance of other modules (prosody) of the ASR system. So, in tonal or pitch-accented languages (such as Chinese), dialect-specific prosodic models [18] are developed. Huge mismatch in vocabulary and grammar can be observed in Arabic dialects [19], dialect-specific language models have shown promising results. In Indian languages, different sandhi rules were established across dialects forming different words.

Considering the acoustic model of the ASR system, the inclusion of dialectal information seems to be promising [9, 10, 12, 15, 17]. Therefore, dialect embeddings from improved dialect classification can

be included in the training of the acoustic model of an ASR system. Dialectal information from improved dialect classifications can also be used in forensic departments or call centres to collect complete information about the speaker or validate the information appropriately. Further, dialect-specific voice assistants can be developed for ease of usage of ASR systems. Custom voice assistants with automatic switching to dialect-specific speech will sound more natural to the listener. This requires automatic identification of dialect from speech.

1.2 Objective and scope of thesis

The article [20] pointed out six existing challenges in speech recognition systems as of 2014. In [11], five current popular commercial speech recognition systems were evaluated, and it was found that the systems are biased towards a group of people compared to others. With the current widespread usage of these devices, this bias or unfair behaviour of these devices for a group of people is unacceptable. So, this thesis proposes approaches that can improve the performance of multi-dialect speech recognition systems.

From the literature, it can be observed that providing dialect information improved the performance. The embeddings which provide dialect information should be efficient to improve automatic speech recognition. So, this thesis aims to improve the performance of a dialect classification system that provides efficient dialect embeddings and discrete dialect classes. Further, the improved dialect embeddings are applied in speech recognition to improve recognition performance.

Figure 1.3 shows the block diagram of a dialect classification system which includes mainly three stages: feature extraction, embedding extraction, and classification. Initial investigations were done with major regional English dialects (AU-Australian, UK-Britain, and US-American). Traditionally, a sequence of feature vectors obtained from blocks of speech signal using the mel frequency cepstral coefficients derived from STFT spectrum (MFCC-STFT) approach is used to represent speech signals for dialect classification. However, the linguistic variations across dialects are present in intricate details of phonetics (aspirations, trills, fricative, etc.) and longer temporal segments (such as phonotactics and prosodic variations). So, advanced signal processing approaches that provide a high degree of spectral resolution (such as single frequency filtering (SFF) and zero-time windowing (ZTW) based features) and signal processing approaches that provide long temporal summarization (such as frequency domain linear prediction (FDLP) based features) are investigated for dialect classification with unsupervised i-vector based approach. Based on the advances and promising performance of deep neural networks in all the applications and also the fact that they can provide long temporal context, different network architectures such as convolution neural network (CNN), temporal convolution neural network (TCN), time-delay neural network (TDNN), and Emphasized channel attention, propagation and aggregation in TDNN (ECAPA-TDNN) that provide different temporal contexts are explored.

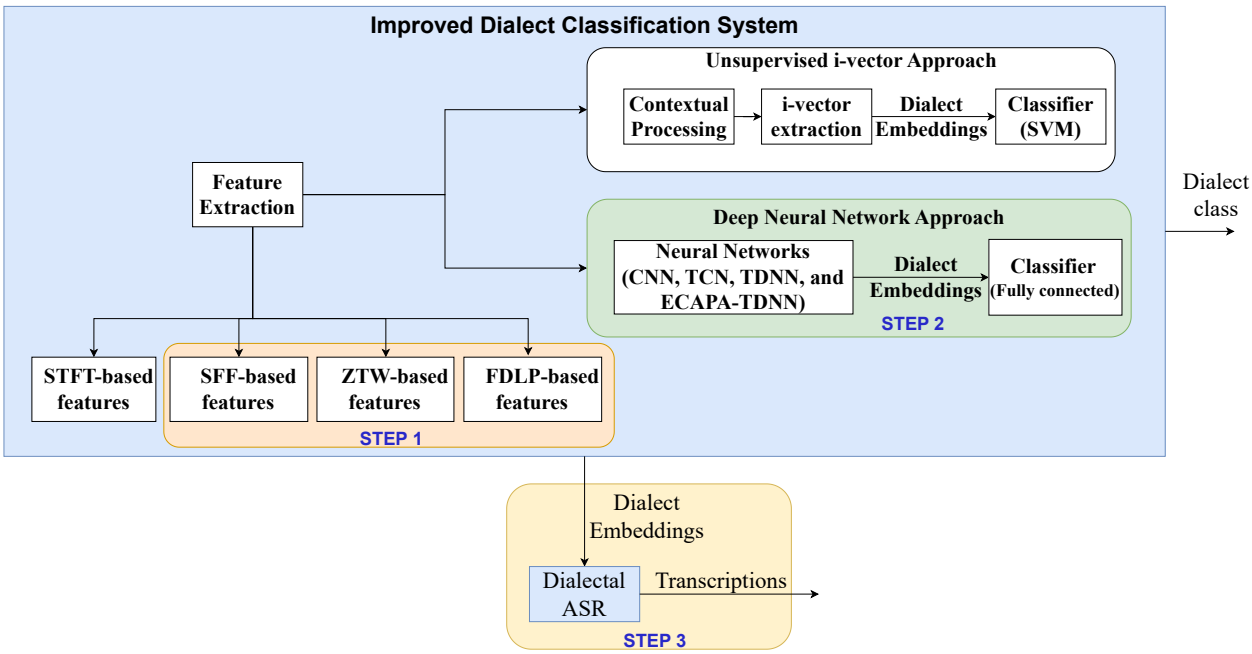


Figure 1.3: Block schematic showing overall flow/scope of the thesis. **Step1:** Proposal of signal processing approaches for feature extraction for dialect classification, **Step2:** Proposal of advanced deep neural network approaches for dialect classification, and **Step3:** Application to multi-dialect speech recognition

Further, these improved dialect embeddings derived from the proposed dialect classification system are applied in the speech recognition system to improve its performance.

The observations from the major dialects of English for dialect classification are extended to Indian English dialects for L1 identification (native language [L1: Hindi, Kannada, Malayalam, Tamil, and Telugu] identification from non-native [L2: English] speech). Finally, the improved dialect embeddings extracted from L1 identification are applied to dialectal speech recognizer to improve the performance in transcription.

1.3 Organization

The thesis is organized as follows: Chapter 2 gives a literature review of studies related to dialect classification and mentions the gaps in the dialect classification system. Chapter 3 describes the UT-Podcast corpus used in the evaluation of dialect classification. Chapters 4-6 uses the traditional i-vector-based dialect classification. Chapter 4 gives the details of the baseline system and evaluates it for dialect classification. Chapter 5 proposes to use features that provide higher spectral and temporal resolution. Chapter 6 proposes to use features that provide higher temporal summarization. Chapter 7 investigates deep neural network-based approaches for dialect classification with proposed features. The embeddings derived

from improved dialect classification are applied in Chapters 8 for automatic speech recognizer (ASR). First, the ASR system with major dialects of English investigates the dialectal embeddings. Then, the proposed better classification approach is used for L1 identification and the L1 embeddings are leveraged in Indian English ASR system. Finally, Chapter 9 summarizes and concludes the thesis. a

Chapter 2

Survey of existing dialect classification systems

Previous studies on the automatic classification of accent/dialect are broadly focused on three areas: first, to find the best frame-level features. Second, to find an approach that appropriately represents the frame-level features, and third, to find the best classifier. Dialect can be varied in speech either in acoustics (distribution of sounds, stress, rhythm, and intonation patterns) [21–24] or phonotactics (sequence of sounds) [25–28] of the speech. Representing the spectral features in unsupervised and compact form is the most popular area of research, where interesting approaches such as i-vectors [21, 22, 29, 30], unsupervised bottleneck features (uBNF) [31, 32], autoencoders with recurrent neural networks [33] and factorized hierarchical variational autoencoder (FHVAE) were explored. The most widely used classifiers are support vector machine (SVM), linear discriminant analysis (LDA) and its variants, such as quadratic discriminative analysis (QDA), probabilistic linear discriminant analysis (PLDA), and heteroscedastic linear discriminant analysis (HLDA) [30, 34–36].

With the introduction of convolution neural networks (CNN) for dialect classification [37, 38] that can extract a compact representation of features along with classification, three stages of classification are reduced to two. In [37], CNNs are evaluated over the Arabic database (MGB-3) with various acoustic features such as mel-frequency cepstral coefficients (MFCC), log mel-scale filterbank energies (FBANK), and spectrogram for dialect classification.

2.1 Feature analysis for dialect discrimination

The factors contributing to discriminating dialects can be categorized into low-level acoustic and high-level linguistic features. Low-level acoustic features are derived directly from the signal without the knowledge of linguistic content. They contribute to the pronunciation or accent variations across dialect classes. High-level linguistic features are dependent on the underlying text of the corresponding speech signal. These features capture linguistic differences such as vocabulary, spelling, and grammar across

dialects. High-level linguistic factors cannot be learned directly from the speech signals, rather need a speech recognition system to extract the linguistic content from the speech signals.

2.1.1 Acoustic analysis for pronunciation variants between dialects

Any sound produced by a human being can be characterized by the pressure built up in the lungs, the vocal fold's vibration, the vocal tract system's shape, the articulator's position and the rate of movement of articulators. Patterns in these characteristics of acoustics discriminate speech against other dialects. From a signal processing point of view, these acoustic variants can be studied in two categories such as spectral and prosodic features.

2.1.1.1 Spectral features

Spectral features can be extracted from speech signals using appropriate parametric representations. These parametric representations should be such that they represent the speech signal compactly, retaining useful information. In 1980, MFCCs were first proposed by Davis, and Mermelstein [39] for speech recognition applications that should replicate the human ear. Later on, any speech application that derives spectral features from speech signals used these parametric representations as state-of-the-art spectral representations.

In the early stages of literature for dialect classification, an accent/dialect-sensitive frequency scale is introduced that contradicts mel-scale representations [2, 40]. This frequency scale worked better than the mel-scale for corpus containing American, Chinese, Turkish, and German accents of English. This study stated that the frequency band between 1.5 kHz - 2.5 kHz is more sensitive to accent discrimination. In [41], linear prediction analysis is used to derive various formant information and concluded that the second formant is the most accent-sensitive format between major dialects of English such as British, American, and Australian. A contradicting study [42] stated that Cantonese non-native English accent can be distinguished by emphasizing more on formant F3. In [43], F1 and F2 show more distinguishable properties, again contradicting the above two studies. The study in [44] looked at different overlapping sub-bands (A: 0 - 0.77 kHz, B: 0.34 - 3.44 kHz, C: 2.23 - 5.25 kHz, and D: 3.40 - 11.02 kHz) to find the appropriate sub-bands for accent identification in contrast to speaker identification. Safavi et al., observed that narrow sub-band region B contains more information with respect to accents in contrast to speaker information over the accents of the British Isles. In one of the recent studies for accent classification with dynamic scale [45], it is observed that the learnt scale is close to non-linear mel-scale.

The speech signal is quasi-periodic, so MFCC features are extracted from the windowed signal. To have a reliable spectral estimate, a window size of $20-40$ msec is considered based on various studies in the

literature. Then each frame is windowed with an overlap of 30-50% using a Hamming window which is proven to reduce leakage.

The dynamic sounds, such as trills and aspirations, which contribute to differentiating dialects, are transient and show the dynamics within a window of 20 msec. Using a 20 - 30 msec window, averages the dynamics of these sounds. Shorter windows could lead to compromise in spectral resolution, while longer window lengths could lose information on some transient sounds.

2.1.1.2 Prosodic features

Prosodic variations between dialects can be observed as rhythm, intonation, duration, and intensity changes. Features related to prosodic variations at these levels are discussed below:

1. **Rhythm:** Rhythm is the periodic pattern in speech signals with an isochrony. Rhythmic patterns can be categorized into three major classes as per Pike [46] and Abercrombie [47]: stress-timed (English and German), syllable-timed (French and Spanish), and mora-timed (Japanese) based on isochrony of unit. Stress-timed languages are characterized to have regular intervals between stressed syllables, syllable-timed languages are periodic with syllables, and successive moras are periodic in mora-timed languages.

   Based on these categories, Ramus et al., [48] investigated various measures such as %V: proportion of vocalic intervals, ΔV: standard deviation of vocalic intervals, and ΔC: standard deviation of consonantal intervals to correlate speech signal to strict categorization. However, languages with intermediate rhythmic patterns exist, too [49]. Despite the discrete categorization of languages, dialects/accents in a language vary continuously in the periodicity of vowels and consonants [50, 51]. For example, native speakers of syllable-timed language exhibit rhythmic characteristics of syllable-timed language when speaking the stress-timed language. Some perception studies observed that listeners distinguished dialects based on their rhythm, hypothesized by these various research methods analyzed rhythm as a discriminative factor for dialects and found cross-dialectal differences using rhythm. Despite English being stated as a stress-timed language, Singapore English is proven to be a syllable-timed dialect of English in [52], and a new measure of rhythm, pairwise variability index (PVI) is introduced in [52] which shows that dialects too vary in rhythm. The metric PVI is widely used and is given by:

$$PVI = \frac{2}{m-1} \sum_{k=1}^{m-1} \frac{|d_k - d_{k+1}|}{d_k + d_{k+1}} \tag{2.1}$$

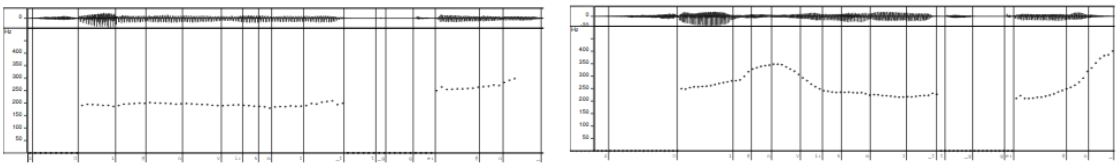
   where d and m are the duration of a vowel and the number of vowels, respectively. By using PVI, in [53], Taiwan English is proven to be a syllable-timed language. [54] shows that PVI between vowels gives distinguishing property than PVI between consonants. In [50], based on the perceptional

studies on Western and Eastern dialects of Arabic, compared vocalic and inter-vocalic distances between dialects. Salem et al. stated in [50] that Western dialects of Arabic are characterized by complex syllables with compressed vowels and Eastern dialects with long vowels. In [24], inclusion of rhythmic features (%V, ΔV, and ΔC) improved the performance of discriminating Arabic dialects.

2. **Intonation:** Intonation is one of the perceptual features which discriminate dialects which can be influenced by a speaker's anatomy, emotion, and type of utterance. It is proved that intonation is one of the discriminating factors for dialect classification in [55] using declarative sentences. Grabe et al., in [56], investigated intonational patterns in accents of British isles (IVIE corpus) with different sentence types and found that dialects differ with intonational topologies defined by ToBI (Tones and Break Indices) labelling.

   A contour of fundamental frequency F0 measures intonation. Methods such as the robust algorithm for pitch extraction (RAPT) and the linear prediction coefficients (LPC) are used to extract pitch. In [57], the pitch of an utterance is computed using RAPT and modelled pitch patterns using pitch codebook and used KL (Kullback–Leibler) divergence to compute the distance between discrete dialect distributions. Using this method on Arabic dialects, the United Arab Emirates dialect is distinguished from Egyptian and Syrian dialects, but Egyptian and Syrian dialects are not distinguishable from each other.

   Figure 2.1 shows the pitch contours of German and Chinese speakers where both speakers utter a question in German. We observe that the pitch rises at the end of the sentence for German, while Chinese speakers exhibited syllable level pitch changes. Peng in [58] used F_0 slope and height to discriminate Mandarin and Cantonese dialects of Chinese, which majorly differ in intones.



(a) The pitch contour of a question from a German speaker (b) The pitch contour of a question from a Chinese speaker

Figure 2.1: Illustration of intonational variations for sentence, "Soll ich mitgehen (Should I come with you?)" (spoken in German) by German and Chinese speaker [1].

In [59], investigated the importance of glottal signal for dialect identification by performing glottal waveform transformation over a Peruvian speaker with Cuban dialect and found that the speech sounded more like Cuban dialect. This shows that intonation is one of the features of discriminating dialects.

3. **Duration**

   Arslan and Hansen in [2] analyzed word-final stop closure duration, voice onset time, average voicing duration, and average word duration for native English speakers and non-native English speakers with Mandarin, German, and Turkish as a native language. Figure 2.2 illustrates the word-final stop closure of the word "would". The duration of stop closure is significantly longer for Mandarin speakers. From the experiments, they concluded that word-final stop closure duration, average voicing, and word durations as accent sensitive features.

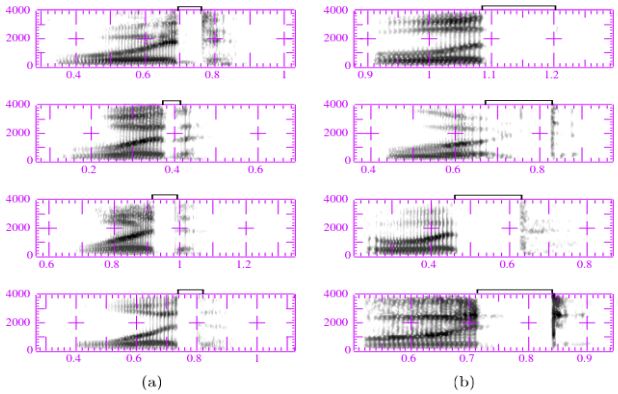


Figure 2.2: Illustration of the change of stop closure durations for the word "would" due to Mandarin accent a) 4 native (German) speakers b) 4 non-native (Mandarin) speakers. [2]

4. **Intensity:**

   Intensity is the loudness of the speech produced by the speaker, and high amplitudes in the signal characterize it. Both intrinsic and extrinsic factors modulate intensity. Intrinsic factors include emotion, dialect, and speaker characteristics, and extrinsic factors include environmental changes such as noise, interrupting speaker, and recording device properties. In [55], recordings are from the same speaker and same environments, which ensures that the only factor which modulates the intensity of speech is accent and proves that intensity is the best feature for discriminating accents of England. But the challenging task is getting the intensity variations that distinguish dialects.

Even though prosody is one of the main factors for identifying dialects from perceptual studies, prosodic features improve the performance but are not the main factor in identifying dialects.

2.1.2 Text-based features

High-level text-based features can capture the phonotactics, lexical choice, and semantic differences in the speech signal across dialects. These text-based features are extracted using a speech recognizer

that decodes the text in the speech signal. These features are widely explored in the literature for dialect classification inherited from language classification.

2.1.2.1 Phonotactics

Dialects in a language tend to add, replace, and substitute new phones. Based on these observations, we can say that the utterances belonging to different dialects have different phone sequences. To capture these phonotactics in speech, phone sequence is used as a clue to identify dialect in previous studies. To model the phonotactics from an utterance, a signal should be processed in two stages. In the first stage, a phone recognizer is deployed to recognize the phones in the utterance, and then in the second stage, these phones are modelled using language models.

Language models can be either the traditional count-based n-gram model or the neural language model. Initial studies on language modelling used n-grams. The posterior probability of a phone is computed from the train data using the following equation:

$$P(ph_i/ph_1, ph_2, \dots) = \frac{count(ph_1, ph_2, \dots, ph_i)}{count(ph_1, ph_2, \dots, ph_{i-1})} \tag{2.2}$$

A phone language model is built corresponding to each dialect and uses perplexity to score the test utterance. The perplexity (PP) is computed as below:

$$PP(ph/\lambda_d) = 2^{H(ph/\lambda_d)} \tag{2.3}$$

where $H(ph/\lambda_d)$ is the entropy of the phone sequence in utterance concerning model λ_d. The lower the perplexity, the closer the utterance to a dialect.

Some of the challenges concerning phonotactic features are: First, it needs a phone recognizer to capture the phonotactic variants in the dialects. Second, the phone labels obtained from these approaches are broad transcriptions, which don't include intricate differences in phone sequences which are highly required for dialect discrimination.

2.1.2.2 Lexical choice and semantics

Word tactics were considered as they are complimentary to acoustic features. Character n-gram and word n-gram are the most popularly used lexical features for dialect identification [60, 61]. For each dialect an N-gram model is trained and represented by $D = \{D1, D2, ...\}$. $P(W/D_i)$ gives the probability of a word sequence belonging to a dialect class and the class with maximum probability is chosen as its predicted dialect. In [61], latent semantic analysis-based features are explored for dialect classification. The dialect is classified based on the minimum cosine distance between the query semantics and dialect semantics.

However, dialectal variations are less exhibited by word tactics when compared between languages. The amount of change in word tactics between dialects depends on the language considered for dialect variations. Word-tactical variations within dialects of English are worse than dialects of Arabic [19]. Among dialects, especially in the English language, change in vocabulary is very low between dialects. It is observed in [61], a very low performance of 35.6% un-weighted recall (UAR) is observed with word-tactics. It can be concluded that information from higher linguistic representations may not be as beneficial for dialect classification, especially for dialects in English.

2.2 Machine learning approaches for learning representations

The overview of all the machine learning approaches for representational learning in the literature applied to dialect classification tasks is discussed in this section.

2.2.1 Generative models for learning latent representation of acoustic features

Generative models are the probabilistic models that play a significant role in machine learning. These models represent the data by their uncertainties using their distribution. Modelling the data using generative models will reduce the data dimensions and generate new data points using the distributions, which can train a generalized model. This generalized model can handle mismatch conditions between train data and real-time data.

Acoustic features are derived from speech signals with respect to each frame of an utterance. A number of acoustic features derived from each utterance are dependent on the duration of an utterance which changes from utterance to utterance. Any statistical classification model which works on an utterance requires fixed dimensional utterance-level features. Therefore, these variable-length acoustic features derived along the temporal axis should be converted to fixed and compact representations and these compact representations should be constrained to contain all the dependencies in the speech signal along the temporal axis. In this section, we discuss the generative models for modelling latent representations used in literature for dialect identification.

2.2.1.1 Gaussian mixture models

The distribution of correlated continuous acoustic features can be modelled by a linear combination of Gaussian components. This is one of the unsupervised approaches to represent the acoustic features based on estimated component distributions. It is hypothesized and shown that these distributions characterize the phoneme distributions. The likelihood of acoustic features is estimated using a linear weighted combination of K Gaussian components.

The modelled distribution of sounds using Gaussian mixture model (GMM) for dialect classification in four variants. They are as Gaussian means, posterior probabilities [62], weights [63], and unsupervised phonetic labelling [31].

2.2.1.2 Factor analysis

Factor analysis is a mathematical model used to decompose correlated attributes into compact and uncorrelated representations. In the dialect identification task, factor analysis is used to decompose both Gaussian mean vectors and Gaussian mixture weights. For decomposing the Gaussian mean vector, the i-vector modelling strategy is used to model and this has been the state-of-the-art until the usage of end-to-end CNNs. Gaussian mixture weights are decomposed and adapted using non-negative matrix analysis.

1. **Factor analysis on Gaussian mean vectors**: The i-vector model is widely used in all speech applications to represent an utterance in shared unobserved uncorrelated factors. Initial studies on other speech applications used joint factor analysis (JFA) for decomposing the Gaussian super vectors. In JFA, two stages of independent factoring are done to extract channel and speaker-related latent features. However, in [64], Dehek et. al., show that the independent latent features share speaker and channel factors. Therefore, i-vectors with a single decomposition are proposed in [65]. Any feature vector can be decomposed into a mean vector (**m**), offset (**Tv**) and a unique component which is not a shared factor with other variables (ε).

$$\mathbf{M} = \mathbf{m} + \mathbf{Tv} + \varepsilon \tag{2.4}$$

   where **m**, **M** are the universal background model (UBM) Gaussian super vectors and maximum a posteriori adaptation (MAP) adapted Gaussian super vectors. The dimension of these supervectors is 1 X $c * f$, where c represents a number of Gaussian components and f represents the feature dimension. **T** is the total variability matrix and v is the unobserved factor which is the i-vector representation with a distribution $\mathcal{N}(0, I)$. ε represents the unique component corresponding to each feature variable.

2. **Factor analysis on Gaussian mixture weights** : Gaussian mixture weights are decomposed using factor analysis and it is represented below [63]:

$$w_c = b_c + L_c r \tag{2.5}$$

   where w_c is the weight of each component. Using factor analysis with a constrained objective function over Gaussian mixture weights the adaptation of weights is done. Below is the optimization problem based on which the weights are adapted.

$$\begin{aligned} &\max_{r} \sum_{t=1}^{\tau} \sum_{c=1}^{C} \gamma_{c,t} log(b_c + L_c r) \\ &\text{subject to } \mathbf{1}(b + Lr) = 1 \text{ Equality constraint} \\ &b + Lr > 0 \text{ Inequality constraint} \end{aligned} \tag{2.6}$$

Unconstrained optimization is faster to converge than constrained optimization. Therefore, the above equation is reformulated with the equality constraint by relaxing the inequality constraint then a simple gradient ascent and projected gradient algorithms are applied to obtain subspace vector r and subspace matrix L_r.

2.2.2 Autoencoders for representation learning

Autoencoder is a neural network architecture that learns to represent latent features while trying to reconstruct the input. It is similar to principal component analysis (PCA), except that they include non-linear activation functions and the weight matrix of the encoder and decoder are not orthogonal to each other. Autoencoder usually has two parts, one trying to find the latent representations (encoder) and the other trying to reconstruct the input features (decoder). The equation of the encoder with one layer in function representation is given below:

$$Z = f(X) \tag{2.7}$$

where the function f takes the form $Wx + b$ with a non-linear activation over it. The decoder in the network tries to generate input features from latent features and its equation is given below:

$$\hat{X} = g(Z) \tag{2.8}$$

where g is the transposed representations of f. Both networks are trained to minimize the distance between X and $\hat{X}$. The objective of the neural network is given below:

$$\begin{aligned} &\min_{\theta} L \\ &\text{where} \\ &L = || X - \hat{X} || \quad \text{and} \end{aligned} \tag{2.9}$$

where θ represents parameters of network. In [33], sequence-to-sequence recurrent neural networks such as (LSTM) and Gated Recurrent Unit (GRU) are used as network architectures in the encoder and decoder framework of autoencoder to learn dialect embeddings.

2.3 Machine learning approaches for the classification of dialects

Both generative and discriminative classification approaches are discussed in this section. GMM is the generative unsupervised approach for classification tasks. SVM and neural models are discriminative supervised approaches for dialect classification.

2.3.1 Gaussian mixture model as classifier

In [66], the Gaussian mixture model (GMM) is used as a classifier for dialect classification. GMM is used to model the distribution of the training data. A class-specific GMM should be trained to model the dialect-specific distributions. But in the case of deficient dialect-specific data, the dialect with deficient data is modelled poorly and won't be a generalized model. Therefore, to overcome this problem, a UBM is trained with complete data comprising all dialects. Next, the MAP estimation [67] of each class distribution is done from UBM. Now, such models are generalized and this approach is faster than training dialect-specific GMM models. Given a test utterance, it is classified based on their inclination towards class-specific GMMs.

2.3.2 Support vector machine as classifier

Support vector machine (SVM) is a very commonly used statistical model for any classification problem. SVM was first introduced by Cortes and Vapnik in 1995 [68]. The main objective of SVM is to learn the hyperplane which optimally classifies the samples in N-dimensional space. Generally, SVM is a two-class classifier. However, to use SVM for multiple class problems either one-vs-all or one-to-one methods have been used. Classifying optimally is to find the best hyperplane out of all the possible hyperplanes i.e., a hyperplane that maintains maximum margin (minimum distance of the hyperplane from the classes). Let us define the hyperplane by a line equation: $w_h^T x + e = 0$, where w_h defines the perpendicular component of the hyperplane. The margin between the hyperplane and the plane passing through the nearest point is given by $\frac{2}{||w_h||}$.

2.3.3 Neural network models as classifiers

Artificial neural network is a framework that attempts to mimic the human brain. Artificial neural network tries to implement a simplified model of neuron simulations in the brain. A very simplified neural network is a perceptron that takes inputs and a single output which is either 0 or 1. Linear or non-linear activation functions are used to model the complexity of data at each node.

2.3.3.1 Feed forward neural network

Feed-forward neural networks are artificial neural networks similar to perceptron. These networks have an input layer, an output layer, and hidden layers. Each layer has multiple nodes that act like neurons in the brain connected to nodes in the other layers. In a feed-forward neural network, all the nodes in one layer are connected to all the nodes in the other. The output of each node is modelled as below:

$$y(\mathbf{x};\mathbf{W},\mathbf{b}) = F(\mathbf{W}^T\mathbf{x}+\mathbf{b}) \tag{2.10}$$

where F is the activation function from each node, the most commonly used activation functions in the literature are sigmoid, tanh, and rectified linear unit (ReLU).

The output layer of the neural network should use an activation function based on classification or regression tasks. The error computed at the output layer is backpropagated along with the derivative of the activation function. The weights at each connection are updated based on the error backpropagated.

2.3.3.2 Convolution neural network

Convolutional neural networks (CNN) are widely used deep network architecture [69, 70], due to their automatic detection of essential features. Its architecture is mainly motivated by the observations in [71], based on the experiments on the visual cortex of a cat. The primary motivation behind this architecture is, it explores the spatial structure in the image and temporal structure in the speech signal. The main advantage of CNN networks over other networks is, it automatically selects the essential features along the temporal axis. In this network, a filter F is convolved with a speech signal by striding along the temporal axis.

2.4 Significant gaps in dialect classification

1. The identification of dialects is a challenging task when compared to language identification due to their highly overlapping phonemic inventory. Most of the models in the literature for dialect identification are borrowed from the language identification problem. There is a need to separately work on dialect identifications.

2. MFCC features derived from STFT were used commonly, these features are derived for every 20-30 msec window. However, the dynamic nature of sounds contributing to dialect discrimination is lost due to windowing over speech signals.

3. Unsupervised i-vector approach was commonly used for dialect classification. However, the factor analysis-based i-vector approach computes all the utterance variations that include speaker characteristics and environmental conditions. Disentangling speaker characteristics from accent features for dialect classification should be explored and applied in automatic dialect classification.

4. Dialects are better classified with higher temporal context. The computational complexity increases by increasing temporal context in CNN.

5. Dialects in each language have different characteristics. In [40, 42], different frequency scales were proposed for dialects in different languages. Therefore, there is a need for language invariant frequency scale which is learned dynamically to identify dialect.

Chapter 3

English speech corpus for dialect classification

Humans communicate using different languages such as English, Spanish, French, etc. These languages are subdivided based on regional, cultural, and social differences into dialects. This thesis considered dialects of English. English has major dialects such as American, British, Australian, and so on. Further, they are divided into many subcategories, such as American into Boston, New Jersey, Texas, etc. and British into Belfast, Bradford, Cardiff and so on. But this thesis considered the classification of speech into major dialects of English such as American, British, and Australian. For the investigation of dialect classification, openly available UT-Podcast [72] corpus is considered across the thesis. The corpus consists of three broad dialects of English: AU (Australian), UK (Britain), and US (American). Within a region (either US, UK, or AU), sub-variants can exist, but as per this corpus, only the primary dialect of the speaker is provided.

UT-Podcast corpus is collected by crawling web-based podcasts that mainly contain interviews. The speech produced is spontaneous and not structured as taken from the interview. Since it is spontaneous, variations can be observed at different levels, such as pronunciation, vocabulary, and grammar. These variations are mainly due to regional differences. These interviews covered news, science, religion, society and life. For a better representation of data, it is collected from a wide range of websites (AU from 12, UK from 5, and US from 8 websites). The speech collected has a sampling frequency of 8 kHz. The duration of each conversation in UT-Podcast ranges from 30 to 60 minutes. These conversations undergo segmentation into smaller utterances through voice activity detection (VAD) to prevent abrupt truncations, as mentioned in [23]. Upon manual inspection of a randomly selected set of audio files, it was observed that very few files exhibited cross-talks. To address this, we performed additional cleaning using pyannote VAD. Subsequently, approximately 10 to 15 samples were selected from each class for further manual verification. The pre-processed speech segments have an average length of about 17 *sec* and 46 words.

After pre-processing and segmentations, there are 1762 utterances in total. Train and test from entirely different websites to have generalized test conditions. After train and test division, there are 1101 utterances in the training set and 661 utterances in the test set. Number of utterances available for training in each of the dialect are, AU:449, UK:246, and US:406. Number of utterances available for test set in each of the

dialect are, AU:332, UK:89, and US:240. In both training and test sets, number of utterance in UK class is low.

Total duration of speech in UT-Podcast is 8.4 hrs. Duration of speech used in train set is 5.2 hrs with 2.1 hrs of AU, 1.2 hrs of UK, and 1.9 hrs of US. Duration of speech used in the test set is 3.2 hrs with 1.6 hrs of AU, 0.4 hrs of UK, and 1.2 hrs of US. Table 3.1 shows the distribution of number of utterances and duration of UT-Podcast corpus across train and test sets of each class. Data was collected from adults, with 127 male and 104 female speakers.

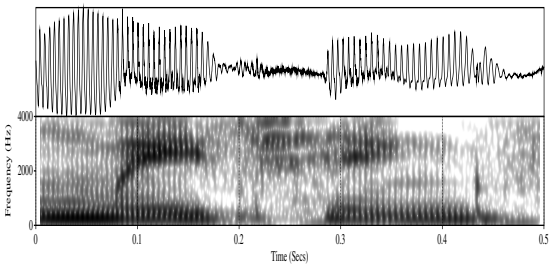
Table 3.1: Distribution of number of utterances (#utterances), the duration of utteraces (#duration (in hrs)), vocab. size, and average sentence length (in words) for each dialect class of UT-Podcast (AU: Australian English, UK: Britain English, and US: American English).

UT-Podcast	**#Utterances**			**#Duration in hrs**			**Vocab size**			**Average sentence len.**		
Data type	AU	UK	US	AU	UK	US	AU	UK	US	AU	UK	US
Train	449	246	406	2.1	1.2	1.9	3923	2025	3224	49.5	48.2	45.4
Test	332	89	240	1.6	0.4	1.2	3178	917	2337	50.5	38.0	45.2

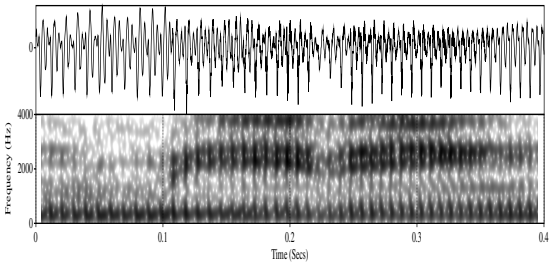
After carefully listening to the audio in UT-Podcast, it is observed that the speech across dialects varies in pronunciation and vocabulary. Pronunciation variations can also be called accent variations. Accent/pronunciation variations can be observed as phonetic replacements or deletions and prosodic variations. One example for phonetic replacements is usage of /d/ by US dialect in words like better, meeting while /t/ is being used by UK dialect. Figure 3.1 shows speech signal and STFT spectrograms for the word "meeting" in UK and US dialects. It can be observed that UK's unvoiced alveolar plosive (/t/) is replaced by voiced alveolar plosive (/d/).

One example of phonetic deletion is due to rhotic vs non-rhotic accents. Figure 3.2, shows the spectrogram for word "better" for three dialects. AU and UK dialects are non-rhotic, while US dialect is rhotic. However, it can be observed that the presence of alveolar flap /ɾ/ cannot be seen clearly in the spectrogram due to its transient/dynamic characteristics. This required better spectrograms that provide better temporal resolution.

Prosodic variations across dialects result in a change in stress, duration, and intonations. Changes in stress results in energy variations in speech across dialects; for example, consider the French loan words such as adult and weekend, where first-syllable stress is observed in the UK dialect while second-syllable stress in the US dialect. The sound length produced changes across dialects; for example, the word "need" has a longer /i/ in the UK than in the US dialect. Figure 3.3 shows the durational variations between US and UK dialects. It can be observed that the UK dialect (0.44 sec) has a longer /i/ sound compared to the US

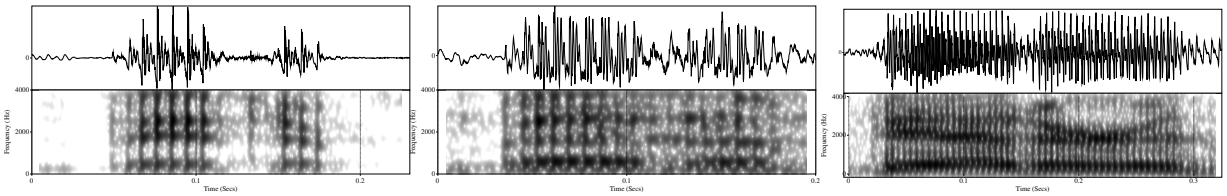


(a) STFT Spectrogram of "meeting" by **UK dialect**



(b) STFT Spectrogram of "meeting" by **US dialect**

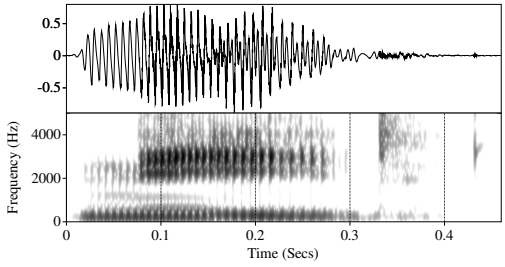
Figure 3.1: Illustration of phonetic replacement for the word "meeting" by British (UK) and American (US) speakers using STFT spectrograms.



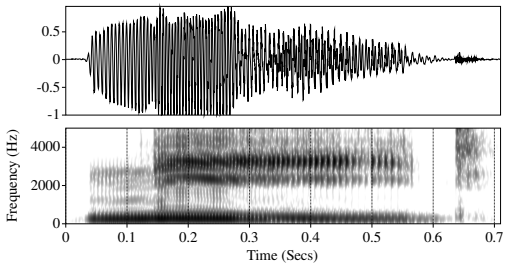
(a) AU (b) UK (c) US

Figure 3.2: Illustration of rhotic variations using word "better" using STFT spectrogram. Each sub-figure represents a dialect, sub-figure (a) represents the Australian (AU) dialect, sub-figure (b) represents the British (UK) dialect, and sub-figure (c) represents the American (US) dialect.

dialect (0.22 secs). The energy and pitch vary across sentences in between dialects. These variations were also observed in the corpus.



(a) STFT Spectrogram of "need" by **UK dialect**



(b) STFT Spectrogram of "need" by **US dialect**

Figure 3.3: Illustration of durational variations using STFT spectrogram for word "need" spoken by British speaker and American speaker.

Also, differences in the usage of vocabulary are observed across dialects. For example, words like "truck" and "apartment" were used in the US, while "lorry" and "flat" were used in the UK. Grammatical variations can also be observed, it is observed that for the past tense of get, the US dialect still used "gotten" while the UK dialect only used "got".

Speech is clear in most of the cases. Very few utterances of podcasts have multiple speakers, with a question from one speaker usually at the beginning of the sentence. So, when considering truncating the speech, it was truncated to the latter part of the speech rather than the initial.

Chapter 4

Dialect classification system: state-of-the-art

Speech in a language can vary in pronunciation, vocabulary, and grammar based on the geographical spread. These systematic variations in speech due to regional diffusion are termed as dialect. Determining the dialect of the speaker from the speech signal is the dialect classification problem. The applications of automatic dialect classification include personalized computer assistant which adapts to user's dialect. Also, the dialect information can be used to improve the performance of automatic speech recognition (ASR) and speaker recognition systems [73, 74]. The origin of the speaker can be determined by dialect classification and this information is useful for profiling and forensics [75].

Dialect classification is similar to language identification, however, the distribution of phones and allophones across dialects is relatively smaller than across languages. This makes dialect classification rather more challenging. Majority of the methods for dialect classification are borrowed from language identification [22, 25, 26, 76]. Previous studies on dialect classification can be categorized into two areas: Studies in first category focused on dialect discriminant feature extraction from speech signal. For example, studies such as [77–79] were focused on exploring the temporal and spectral characteristics across dialects. The features can be further categorized into two; i.e., acoustic or phonotactic based features. Acoustic features usually represent characteristics of speech signal in time or spectral domain, while phonotactic-based [25–28] features are discrete and capture the distribution of phoneme sequences. In [35], the characteristics of sound sequence are captured from the spectral features using stochastic trajectory model. For acoustic-based features, static Mel frequency cepstral coefficients (MFCC) along with shifted delta cepstral (SDC) features of MFCC are widely used [22, 80]. Figure 4.1 shows block schematic diagram of dialect classification system. Based on the acoustic characteristics, the dialect of speech signal is identified. It involves four stages, first extraction of features from speech signal, second contextual processing of frame-level features, third back-end pre-processing to obtain utterance level features, and in final step classifier classifies the dialects.

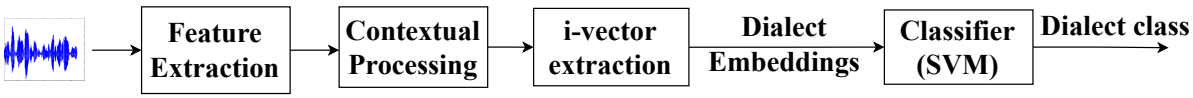


Figure 4.1: Block diagram showing i-vector based baseline dialect classification system.

4.1 Feature extraction methods

This section discusses two different baseline feature extraction methods (MFCC-STFT and LPCC) as baseline features. MFCCs derived from STFT are commonly used acoustic features for dialect classification.

4.1.1 Mel-frequency cepstral coefficients

Conventional mel-frequency cepstral coefficients (MFCC-STFTs) are widely used in all speech applications and considered as baseline for dialect classification [29, 81–84]. Extraction of MFCC-STFTs is motivated by the physiology of human ear [85], where the mel-spaced filters are used, mimicking the physiology of the human ear. The one-dimensional pre-emphasized speech signal is segmented into shorter sliding windows and transformed to spectro-temporal representation (spectrogram) using Fourier transformation. Mel-spaced triangular filter-banks are integrated with the STFT power spectrum along the frequency axis to obtain STFT-based mel filter bank energies. This process integrates higher dimensional time-frequency representation to lower dimensional representations by averaging in spectral sub-bands.

Discrete cosine transformation (DCT) is applied over the log of STFT-based mel filter bank energies and the resultant cepstrum value in each time-channel bin. The cepstral analysis separates the vocal-tract system characteristics into lower order coefficients and excitation source characteristics into higher order cepstral coefficients. Cepstral coefficients are truncated to different cepstral orders for investigating the effect of cepstral order on dialect classification.

4.1.2 Linear prediction cepstral coefficients

Linear prediction analysis is an all-pole based approach approximating the smoothed power spectrum [86]. The spectrum estimated using linear prediction analysis is called linear prediction (LP) spectrum. LP spectrum is a smoothened version of STFT-based power spectrum as it highlights high energy components such as formants and degrading low energy harmonics in the spectrum. The cepstral coefficients derived from the LP spectrum are called linear prediction cepstral coefficients (LPCCs). LPCCs were investigated in speech recognition as they better represent the vocal-tract information [87], language identification [88], and speaker recognition [89].

LP analysis states that any sample point in a signal can be estimated as the linear weighted sum of past samples of the signal. Linear prediction coefficients (LPCs) are estimated by minimizing the mean

square error between the estimated and actual signal. Auto-correlation coefficients are computed from the segmented speech. Finally, LPCs are converted to the cepstral domain to yield LPCCs.

4.2 Contextual processing approaches

Speech signal is assumed to be stationary in 20 - 30 msec window, so features are computed for every 20 - 30 msec frame. Every frame contains information only within 20-30ms. For dialect classification, even the articulatory information is also essential. To capture co-articulatory information between frames, different contextual approaches were explored in this chapter. These approaches include context at frame level by adding delta coefficients. Two main contextual processing approaches, such as delta and double delta coefficients and shifted delta cepstral coefficients, are discussed in this chapter.

4.2.1 Delta and double delta coefficients

The co-articulatory effects in speech results in intra-dependencies across the frames. The co-articulation effects differ from one dialect to other. This co-articulation cannot be captured in static cepstral coefficients therefore delta and double delta (Δ+ΔΔ) cepstral coefficients can be used to capture this effect. Δ+ΔΔ coefficients [90] are computed as follows:

$$\mathbf{d}[\mathbf{t}] = \mathbf{c}[\mathbf{t}+\mathbf{m}] - \mathbf{c}[\mathbf{t}-\mathbf{m}], \tag{4.1}$$

where t defines the current frame index and m defines the context length which is set to 3 in this study. The delta and double delta (Δ+ΔΔ) coefficients compute the delta operation over the delta coefficients.

4.2.2 Shifted delta cepstral coefficients

In [80], it was shown that cepstral features vary temporally across dialects. There was a significant improvement in language identification after using SDC features rather than delta and double delta coefficients [80]. SDC features are computed over the ZTWCCs for each frame. $N-d-p-K$ defines the configuration for the SDC computations. At every time instant t, delta computations between cepstral coefficients at $(t+ip-d)^{th}$ and $(t+ip+d)^{th}$ are done. These delta coefficients computed with i varying from 1 to K, and are stacked to get delta coefficients at each instant in time t. SDC vector $\Delta\mathbf{c}(\mathbf{t},\mathbf{i})$ for cepstral coefficients at time t for i^{th} shift is given by:

$$\Delta\mathbf{c}(\mathbf{t},\mathbf{i}) = \mathbf{c}(\mathbf{t}+\mathbf{ip}+\mathbf{d}) - \mathbf{c}(\mathbf{t}+\mathbf{ip}-\mathbf{d}), \tag{4.2}$$

where N denotes dimension of static cepstral coefficients, d denotes delay or advance from the current frame, p is the shift between consecutive delta computations, and K such delta computations are concatenated to form $N*K$ dimensional SDC coefficients.

4.3 Back end preprocessing (i-vector extraction) approach

Factor analysis is a method of expressing the variability of the observed variables (data) in terms of low-dimensional (latent) vectors. I-vector modeling is one of the factor analysis methods to represent low-dimensional total variability factors for each utterance in a single vector [64]. Stacked means of GMM are termed as supervectors. supervectors of a GMM-UBM are represented by $\mathbf{m}$ and supervectors of an utterance adapted GMM are represented by $\mathbf{M}$. The supervectors of each utterance $\mathbf{M}$ can be expressed by mean components and offset which is given by:

$$\mathbf{M} = \mathbf{m} + \mathbf{T}\mathbf{w}, \tag{4.3}$$

where $\mathbf{T}$ represents low-rank total variability matrix, $\mathbf{w}$ is utterance specific latent factor vector known as i-vector with a prior distribution of $\mathscr{N}(0, \mathbf{I})$.

Means and variances of GMM-UBM were initialized using k-means clustering, then UBM is trained using expectation-maximization (EM) algorithm with train data from all dialects. To obtain i-vectors, a process similar to estimation of Eigen voice in [91, 92] is followed. In this approach, first Baum-Welch statistics per utterance are accumulated then total variability matrix $\mathbf{T}$ is iteratively trained. Finally, i-vector $\mathbf{w}$ is estimated for each utterance which can be used for classifying dialects. Both the GMM-UBM and the total variability matrix $\mathbf{T}$ are trained for five iterations.

4.4 Classification methods

Classifier is trained with utterance level i-vectors to classify dialects. The SVM was trained with a linear kernel in one-vs-rest fashion. The standard publicly available implementation of SVM [93] is leveraged.

4.5 Experimental Setup

This section gives the details of evaluation metrics and configurations of baseline features and baseline dialect classification system.

4.5.1 Configuration of baseline feature extraction

The most popular and conventional Mel-frequency cepstral coefficients (MFCCs) and linear prediction cepstral coefficients (LPCCs) are considered baseline features for dialect classification in this thesis. For all the feature extraction, a window size of 25 msec and half of the window length are considered window shifts. Autocorrelation formulation is used in the extraction of LPCC features. The baseline feature representations are investigated by varying the number of static cepstral coefficients (from 13 to 60).

From static coefficients, Δ, ΔΔ, and SDC coefficients [80] are also derived, which are also investigated to see their effectiveness on dialect classification. For Δ and ΔΔ computation, a delta of one, leading to a context of three is considered. For SDCs, a standard configuration of N-d-p-K = $N-1-3-7$ is considered, where N denotes the dimension of the static cepstral coefficients, d denotes the delay/advance from the present frame; p is the shift between consecutive delta computations; and K such delta computations are concatenated to form N×K-dimensional SDC coefficients.

4.5.2 Configuration of i-vector system

Extraction of i-vectors is motivated by the factor analysis modelling, where features are represented in terms of uncorrelated components [64]. In this, GMM-UBM (trained on all utterances) model is adapted to represent a variable length utterance in terms of fixed representation called super-vectors. Later by the factor analysis, super-vectors are further compressed to retain only an uncorrelated low-dimensional components of super-vectors, which are called i-vectors. Adapted super-vector $\mathbf{m}$ can be represented as $\mathbf{m} = \mathbf{M} + \mathbf{Tv}$; where $\mathbf{M}$ represent mean super-vector obtained by training GMM-UBM with features from all dialects, $\mathbf{T}$ represents total-variability matrix and $\mathbf{v}$ represents i-vectors. The means and variances of GMM-UBM are initialized using k-means clustering. Initial experiments were conducted by varying number of Gaussian components (256, 512, 640, and 1024) with i-vector system trained with MFCC features. From the experiments, it was observed that 640 Gaussian components performed better than all others and hence the number of Gaussian components is set to 640 across all the experiments. GMM is trained with all the dialects to obtain means of GMM-UBM model (represented by M) from the pre-initialized means and variances using k-means clustering. Then the means of GMM-UBM are adapted to each dialect class (represented by m). Factor analysis model is trained for 5 epochs to learn the total variability matrix (represented by T) using Baum welch statistics. From means (m and M) and learnt total variability matrix (T), 100-dimensional i-vectors are computed for each utterance. More details about i-vector extraction can be found in [91, 92]. Matlab toolbox [1] is used for implementing i-vector framework [94]. Similar configuration is being used in a future chapters with proposed features for a fair comparison.

4.5.3 Evaluation metric

The corpus UT-Podcast has imbalanced classes in test set. For classification tasks usually accuracy is evaluation metric, and accuracy is defined as (#correct predictions)/(#total samples). With the imbalanced classed in corpus, the overall accuracy depends on the accuracy of the majority class. While UAR is unweighted average recall which tries to give equal weight to each class irrespective of their strength [95]. So, UAR is considered as primary metric across this thesis.

[1] https://github.com/wangwei2009/MSR-Identity-Toolkit-v1.0

In 2 class classification problem, UAR is expressed as $\frac{(sensitivity+specificity)}{2}$, where $sensitivity = \frac{TP}{(TP+FN)}$ and $specificity = \frac{TN}{(TN+FP)}$. Sensitivity is same as recall/accuracy of positive class and specificity is same as recall/accuracy of negative class.

In case of multi-class classification problem with more than 2 classes, recall (REC) can be expressed as:

$$REC_i = \frac{TP_i}{TP_i + FN_i} \tag{4.4}$$

or

$$REC_i = \frac{\#correct\ predictions\ of\ class_i}{\#total\ samples\ in\ class_i} \tag{4.5}$$

$$UAR = \frac{\sum_1^C REC_i}{C} \tag{4.6}$$

where TP is true positives, FN is false negatives, and REC is Recall.

Along with UAR, class-wise accuracies and confusion matrices are also reported for discussions.

4.6 Results and discussion

This section reports the results and discusses them with baseline approaches (MFCC-STFT with i-vectors and LPCC with i-vectors). In the i-vectors computation, number of GMM components were analyzed with MFCC-STFT static+Δ+ΔΔ. Then, to understand the importance of temporal context and to find the best configurations, different contextual processing approaches (static, static+Δ, static+Δ+ΔΔ, and static+SD coefficients) were investigated.

4.6.1 Hyperparameters tuning for i-vector system

Table 4.1 shows the performances of i-vector-based dialect classification with different GMM components (256, 512, 640, 1024). It is observed that with 640 GMM components, the performance of dialect classification is better. Therefore, 640 is considered in further experiments.

Table 4.1: Performance (in UAR%) for dialect classification with different number of GMM components (256, 512, 640, and 1024) used in i-vector extraction.

No. of GMM components	256	512	640	1024
MFCC+Δ+ΔΔ	72.9	72.6	**74.5**	73.4

4.6.2 Comparison of baseline methods

Table 4.2 shows the performance (in UAR%) for i-vector based dialect classification system with all different temporal contexts such as static, static+Δ, static+Δ+ΔΔ, and static+SD coefficients. With MFCC-STFT features (row 3 of the table), it can be observed that increasing the temporal context consistently improved the performance of MFCC-STFT. However, with LPCC features (row 5 of the table), it can be observed that it improved context only with static+Δ+ΔΔ coefficients. Comparing both MFCC-STFT and LPCC, it can be observed that MFCC-STFT outperformed with a UAR of 77.98%.

Table 4.2: Performance (in UAR%) for i-vector based dialect classification system with all different temporal contexts such as static, static+Δ, static+Δ+ΔΔ, and static+SDC coefficients with baseline features MFCC-STFT and LPCCs.

Feat. type	**static**	**static+Δ**	**static+Δ+ΔΔ**	**static+SDCs**
MFCC-STFT	75.67	76.38	77.21	**77.98**
Rel. Imp.	-	0.94	2.04	3.05
LPCC	69.68	69.57	**74.42**	71.21
Rel. Imp.	-	-0.16	6.80	2.20

4.6.3 Analysis using confusion matrices

Further to understand the class-wise accuracies with confusion of classes, confusion matrices are reported in Figure 4.2. It can be observed from both the figures that the samples of UK class are confused with AU and US. Comparing class-wise performances of STFT-MFCC and LPCC, it can be observed that STFT-MFCC performed better for all the classes, especially the most confused UK class.

4.6.4 Comparison to previous studies

This section compares the results obtained in the current study (i-vectors derived baseline features with SVM) with the previous studies [72] with i-vector based approach. In [72], both text based and audio based approaches were investigated.

In text based approach, term-frequency and inverse document frequency (TF-IDF) was exploited. TF-IDF measures the originality of word in a document. In audio based approach, GMM super-vectors and i-vectors were used with SVM classifier. A fusion of both text and audio approach is also investigated. Among all the approaches in previous studies, i-vector based approach performed better with a UAR of

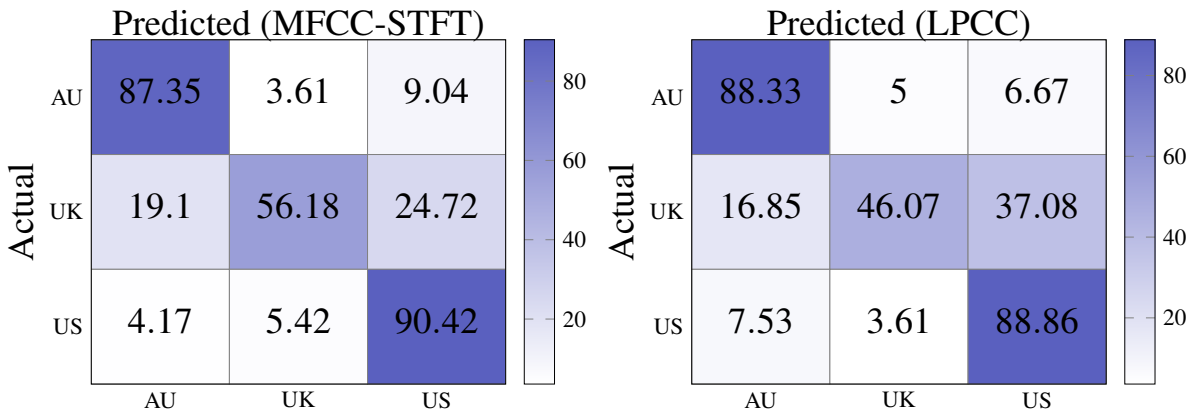


Figure 4.2: Confusion matrices for i-vector based dialect classification system with static MFCC-STFT+SDCs and static LPCC+Δ+ΔΔ .

74.5%. It is also observed that acoustic information is more helpful in identifying dialects when compared to text information.

Table 4.3 shows the performances in UAR% and class-wise accuracies of previous (rows 3-6) and current studies (rows 8-9). Comparing current studies to previous studies, it can be observed that MFCC-STFT with a better configuration of i-vectors performed better than the previous study's i-vector system.

Table 4.3: Comparison of current baseline i-vector system with previous dialect classification models over UT-Podcast corpus (in UAR% and class-wise accuracies).

Arch. type	**UAR**	AU	UK	US
Text and audio based approaches from previous studies [72]				
Audio System (GMM)	**60.3**	85.5	32.6	62.9
Audio System (i-vector)	**74.5**	78.0	61.8	83.8
Text System (TF-IDF logistic regression)	**58.7**	83.1	32.6	60.4
Audio-Text system (Fusion)	**76.3**	86.1	60.7	82.1
i-vector system (current study)				
MFCC-STFT (baseline)	**77.98**	87.35	56.18	90.42
LPCC (baseline)	**74.42**	88.33	46.07	88.86

4.7 Summary and conclusions

Among the baseline features, MFCC-STFTs performed better. On investigation of features with different cepstral order, the impact of cepstral order on the performance of dialect classification differed for each feature representation. Similarly, an investigation of features with different temporal contexts is done, and the impact of temporal context on the performance of dialect classification differed for each feature representation. From this, we can conclude that features behave differently in different temporal contexts.

The short-time windowing of MFCC-STFT will have only a context of 25 msec. The windowing averages the information in each window of 25 ms. The dynamic sounds, such as trills and aspirations, differentiate dialects, are transient. To capture these dynamics, shorter windows are required. However, shorter windows could compromise spectral resolution, while longer window lengths could lose information about some transient sounds. It is also observed that to discriminate dialects, we need a longer temporal context. So, we hypothesize that features that provide higher temporal resolution and temporal context perform well for dialect classification.

Chapter 5

SFF-based and ZTW-based approaches for dialect classification

Dialectal variations can be observed at sub-segmental, segmental, supra-segmental, and sentence levels. The short-time Fourier transform (STFT) features are extracted from the spectrum estimated from every 25 msec framed signal. This averaging within a frame of 25 msec leads to a loss of information related to transient or dynamic sounds. The characteristics of these transient or dynamic sounds vary between dialects. So, this chapter proposes the features derived from advanced signal processing approaches, namely, single frequency filtering (SFF) and zero-time windowing (ZTW) approaches. The spectrum is extracted at every sample for SFF and ZTW methods. This captures the intrinsic variations between dialects.

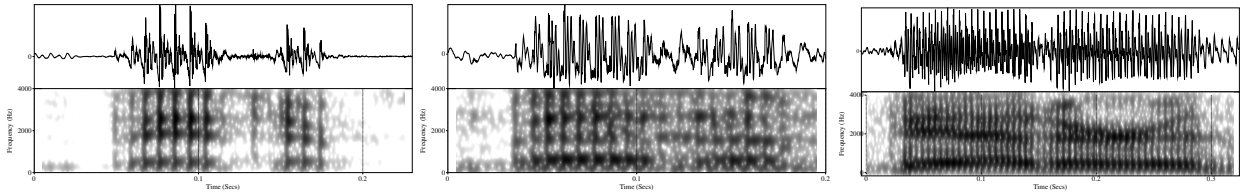
Further, from the previous studies, it is observed that the spectrum computed by SFF has been shown to give good spectral resolution to indicate harmonics and resonances [96]. It has also been observed to give good temporal resolution to model speech excitation features such as impulse-like events [97]. The SFF spectrum has shown promising performance in determining burst-onset points (related to voice-onset time (VOT)) and glottal closure instances compared to the STFT spectrum [97–99]. Previous studies in dialect identification have shown the significance of VOT for the identification of accent [100]. Inspired by this, we propose to use the SFF-based features for dialect identification.

Zero-time windowing method that can effectively differentiate different speech sound characteristics compared to the DFT spectrum [101–103]. In [21], spectral features in the i-vector approach were replaced by speech attributes such as manner and place of articulation. This approach has reduced the relative error rate significantly as compared to MFCC i-vector-based approach. From this, we hypothesize that ZTW-based features might provide better dialect discrimination.

5.1 Motivation for ZTW and SFF methods

Both SFF and ZTW methods compute the spectrum at every sample avoiding the windowing. For illustrations, the word “better” is taken from the UT-Podcast corpus for major English dialects (AU, UK, and US). Due to linguistic phonetic variations between dialects (AU, UK, and US) [7], the word better

is pronounced as /ˈbɛdər/, /ˈbɛtə/, and /ˈbɛdər/ in US, UK, and AU dialects respectively. AU and UK are non-rhotic pronunciations while US is rhotic. US and AU uses voiced alveolar plosive (/d/), while UK uses unvoiced alveolar plosive (/t/). Figure 5.1 shows the STFT plots for the word "better" spoken in AU, UK, and US dialects. It can be observed that the presence of alveolar flap /r/ cannot be seen clearly in the spectrogram due to its transient/dynamic characteristics.



(a) AU (b) UK (c) US

Figure 5.1: Illustrations of rhotic variations using STFT spectrogram for the word "better". Each sub-figure represents a dialect, sub-figure (a) represents the AU dialect, sub-figure (b) represents the UK dialect, and sub-figure (c) represents the US dialect.

To articulate /r/, the tip of the tongue flutters at the alveolar ridge. This flutter causes trill cycles. The sound /r/ can be either produced as a trill or as just a tap. Figure 5.2 shows the pronunciation of trill /r/ in isolation. From the figure, it can be noticed that there are ripples in the amplitude of the ZTW spectrogram due to the presence of secondary excitation introduced by the tongue tip at the alveolar ridge (during the pronunciation of /r/). Trill has atmost around five trill cycles over 200 msec of duration [103]. For 100ms, three cycles of amplitude variations can be seen in the figure (i.e., at a rate of 30Hz). Similar observations can also be seen from the SFF spectrum in Figure 5.3.

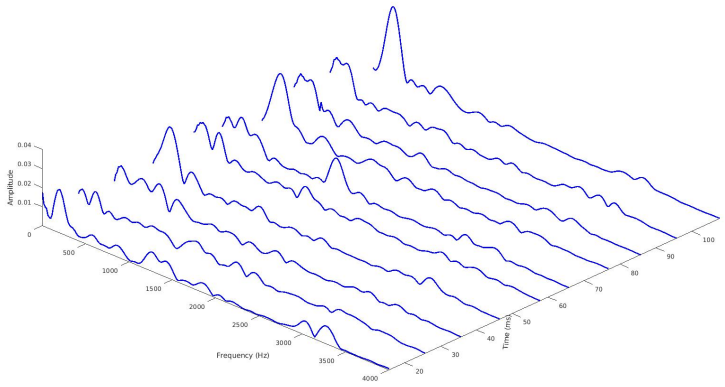


Figure 5.2: Illustration of ZTW spectrum for the sound /r/ at every 10 msec.

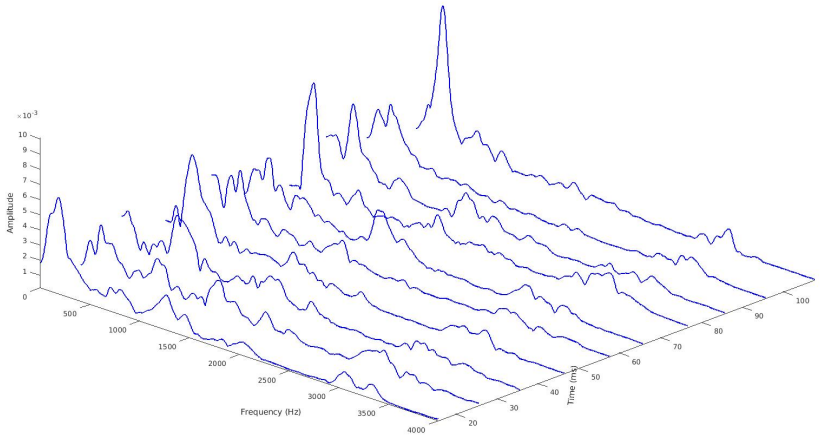


Figure 5.3: Illustration of SFF spectrum for the sound /r/ at every 10 msec.

Figure 5.4 shows the ZTW spectrum for last 40 msec of the word "better" for that has the pronunciation of /r/ by US speaker. From the Figure, it can be noticed that there are ripples in the ZTW spectrum due to the presence of secondary excitation introduced by tongue tip at the alveolar ridge (during pronunciation of /r/). Since the speech is spontaneous and the /r/ occurred as a alveolar tap, only one period is visible in spectrum. Similar observations are also made from SFF spectrum. The temporal resolution of these approaches may help in representation of transient features of speech. With these evidences, we hypothesized that the features derived from SFF and ZTW are better for classification of dialects.

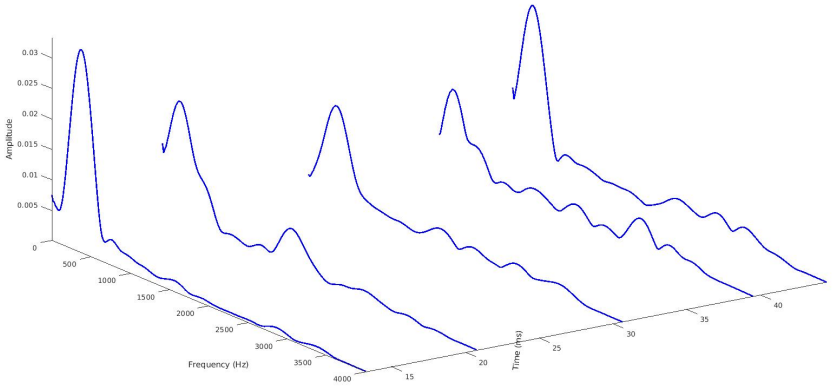
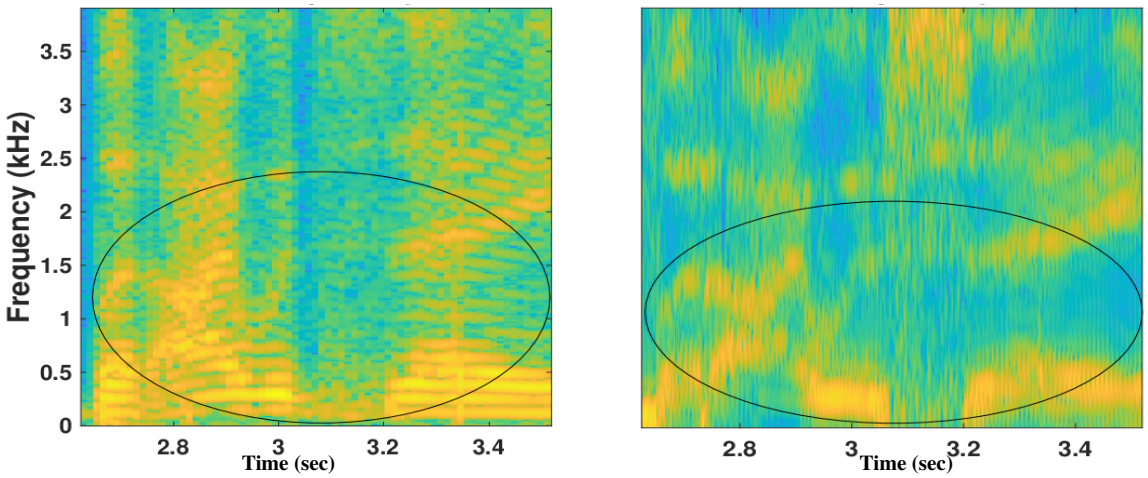


Figure 5.4: Illustration of ZTW spectrum for the sound /r/ in the word "better" taken from US dialect of UT-Podcast corpus.

5.2 ZTW based features

The high resolution spectrum provided by the zero-time windowing method can differentiate different speech sound characteristics effectively compared to the DFT spectrum [101–103]. Previous studies in dialect classification [21] has shown the significance of manner of articulation and place of articulation for dialect classification. This approach reduced the relative error rate significantly as compared to MFCC i-vector based approach showing the importance of formant locations. ZTW method was proposed in [101] to derive the instantaneous spectral characteristics, so that the time-varying characteristics of speech production mechanism can be captured.

Illustrations of spectrograms obtained with STFT and ZTW methods are shown in Figures 5.5(a) and (b) respectively. From the figures, it can be clearly seen that ZTW spectrogram (Figure 5.5(b)) highlights the formant structure compared to STFT spectrogram (Figure 5.5(a)).



(a) **STFT Spectrogram** (b) **ZTW Spectrogram**

Figure 5.5: Illustration of spectrograms obtained with (a) STFT and (b) ZTW methods.

This section first describes the ZTW method used for deriving high−resolution spectrum [104], and then gives a procedure to extract the proposed features from ZTW spectrum.

5.2.1 ZTW method

The instantaneous spectral characteristics of the ZTW spectrum lead to a better representation of time-varying spectral characteristics of speech production mechanism [104]. First, the speech signal is windowed by a heavily decaying window that emphasizes the samples near the start of the window. Spectrum is estimated using group delay at every time instant. ZTW spectrum estimated provides high

temporal resolution (as estimated at every instant) and spectral resolution (due to group delay function). The steps involved in extracting the ZTW spectrum are shown in Figure 5.6 and described as follows:

- At every instant of the speech signal, the segment of L msec (i.e., M samples of $s[n]$, where $M = \frac{Lf_s}{1000}$) is multiplied by a heavily decaying window $w_1^2[n]$. where

$$\begin{aligned} w_1[n] &= 0, && n = 0, \\ &= \frac{1}{4\sin^2(\pi n/2N)}, && n = 1,2,\ldots,N-1. \end{aligned} \tag{5.1}$$

  where N is the number of points used in the computation of DFT ($N >> M$). Multiplying the signal with $w_1^2[n]$ is approximately equivalent to integration in the frequency domain [104]. L=25 msec and N=1024 are chosen for this thesis.

- To reduce the ripple effect in the frequency domain due to segmentation, the signal is multiplied by another window $w_2[n]$, for $n = 0,1,\ldots,M-1$, defined as:

$$w_2[n] = 2(1+\cos(\pi\, n/M)) = 4\, \cos^2(\pi n/2M), \tag{5.2}$$

- The spectrum (X[k]) is estimated from the windowed signal by computing the numerator group delay (NGD). The NGD function is given by:

$$g_n[k] = X_R[k]Y_R[k] + X_I[k]Y_I[k], \qquad k = 0,1,2,\ldots,N-1. \tag{5.3}$$

  where $X_R[k]$ is the real and $X_I[k]$ is imaginary parts of the $X[k]$ (DFT of $x[n]$). Likewise, $Y_R[k]$ is the real and $Y_I[k]$ is the imaginary part of the $Y[k]$ (N-point DFT of $y[n] = nx[n]$).

- NGD spectrum is double-differentiated to highlight the hidden spectral characteristics. The Hilbert envelope of the double-differentiated NGD spectrum is referred to as the ZTW spectrum, denoted by $S[n,k]$.

.

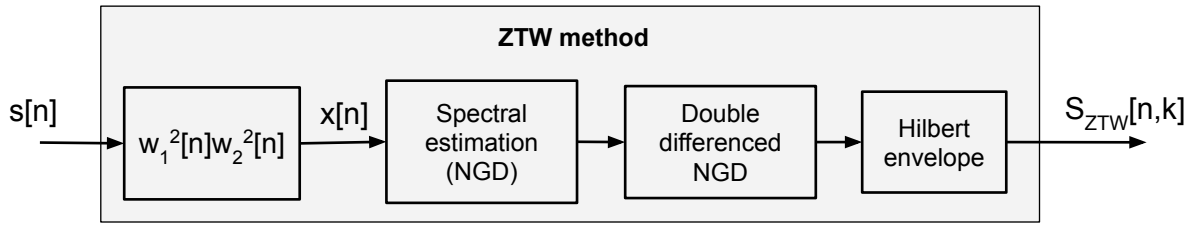


Figure 5.6: Schematic block diagram describing the steps involved in the computation of ZTW spectrum.

5.2.2 Extraction of feature representations from ZTW method

This thesis proposes to derive two types of features from the ZTW spectrum. They are: (1) zero−time windowed cepstral coefficients (referred to as ZTWCCs), (2) mel frequency cepstral coefficients derived

from ZTW spectrum (referred to as MFCC−ZTW). Out of these two, only ZTWCC was investigated for dialect classification in [105].

5.2.2.1 Extraction of zero−time windowed cepstral coefficients (ZTWCCs)

ZTWCCs are computed from the cepstrum of the ZTW spectrum ($S[n,k]$), as follows:

$$C_{ZTW}[n,k] = \text{IFFT}(\log_{10}(S[n,k])). \tag{5.4}$$

From cepstrum, $C_{ZTW}[n,k]$, the first 80 coefficients are considered in this study. The schematic block diagram describing the steps involved in the extraction of ZTWCCs is shown in Figure 5.7.

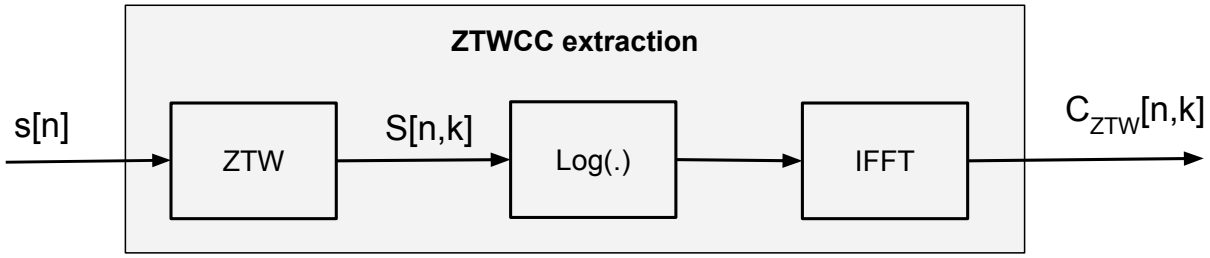


Figure 5.7: Schematic block diagram describing the steps involved in the computation of ZTWCC features from ZTW spectrum.

5.2.2.2 Extraction of MFCCs using the ZTW spectrum (MFCC−ZTWs)

Mel-filter bank energies of the ZTW spectrum are obtained by applying mel filter-bank analysis on the ZTW spectrum. Then, DCT is applied over the log mel-filter bank energies of the ZTW spectrum to obtain MFCC-ZTW features, as follows:

$$MFCC_{ZTW}[n,k] = DCT(\log(Mel(\mid S[n,k] \mid^2))), \tag{5.5}$$

where $MFCC_{ZTW}[n,k]$ denotes the mel-cepstrum. From the mel-cepstrum, all 80 cepstral coefficients (including the zeroth coefficient) are considered. Figure 5.8 is the block diagram showing the extraction process of MFCC-ZTW features from the ZTW spectrum.

5.2.3 Results and discussion

First, this section investigates the different configurations for windowing of the ZTW spectrum. Then, this section provides experimentation results and discusses different cepstral orders for ZTWCC and MFCC-ZTW with i-vector-based dialect classification. Then, it investigates different temporal contexts

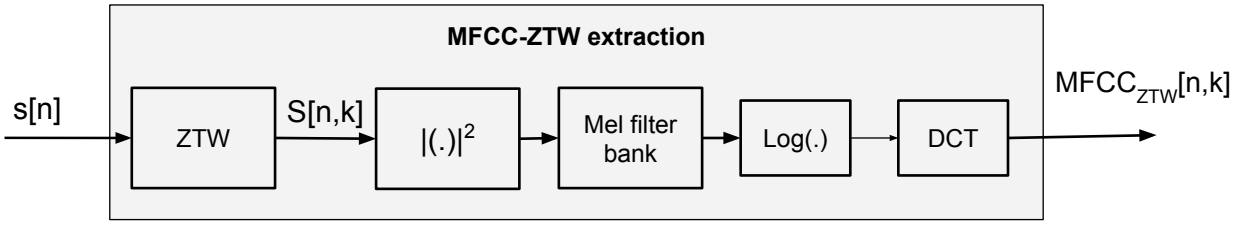


Figure 5.8: Schematic block diagram describing the steps involved in the computation of MFCC-ZTW features from ZTW spectrum.

for ZTW-based features. Finally, we compare the performances of ZTWCC and MFCC-ZTW to baseline MFCC-STFT using confusion matrices for dialect classification.

5.2.3.1 Different segmenting approaches

Instead of considering the ZTW spectrum at each time instant, computational load is reduced by considering the spectrum in a segment of T msec. One of the following four approaches can be used in defining the spectrum using the segment of T msec.

- (a) **Average ZTW spectrum (ZTW$_{avg}$)**: In this approach, the ZTW spectrum is computed by averaging the amplitude envelope $S[n]$ for every frequency k over the entire segment.
- (b) **Minimum ZTW spectrum (ZTW$_{min}$)**: In this approach, the ZTW spectrum is selected as the instantaneous spectrum of $S[n]$, which shows the minimum spectral energy (sum of the squared amplitude envelope values) over the entire segment.
- (c) **Maximum ZTW spectrum (ZTW$_{max}$)**: In this approach, the ZTW spectrum is selected as the instantaneous spectrum of $S[n]$, which shows the maximum spectral energy over the entire segment.
- (d) **Uniform SFF spectrum (ZTW$_{uniform}$)**: In this approach, the SFF spectrum is computed by sampling $S[n]$ at regular intervals defined by the segment duration.

Table 5.1 shows the performance (in UAR%) for i-vector based dialect classification system with static+Δ+$\Delta\Delta$ for ZTWCC and MFCC-ZTW based methods with different strategies (such as average, minimum, maximum, and uniform ZTW spectra). Considering 'l' as the length of utterance in terms of samples, 's' as the number of window frames, and 'n' as the total number of utterances, the performances and the time complexities of extraction of ZTW features were compared.

From table, it can be observed that the time complexities of ZTW$_{avg}$, ZTW$_{min}$, and ZTW$_{max}$ is O(n*l) and the time complexity of ZTW$_{uniform}$ is O(n*s). Here, the number of samples in utterance (l) is much greater

Table 5.1: Performance (in UAR%) for i-vector based dialect classification system with static+Δ+ΔΔ coefficients for ZTW features with different segmenting approaches. The time complexity for each approach is expressed with l as the number of samples in an utterance, s as the number of window frames (l $\ggg$ s), and n as the number of utterances.

Feat. Type	**ZTW$_{avg}$**	**ZTW$_{min}$**	**ZTW$_{max}$**	**ZTW$_{uniform}$**
ZTWCC (static+Δ+ΔΔ)	76.39	72.67	72.55	**78.23**
MFCC-ZTW (static+Δ+ΔΔ)	77.02	74.02	75.84	**78.73**
Time complexity	O(n*l)	O(n*l)	O(n*l)	O(n*s)

than the number of frames (s), with a window slide of 10 msec and 8000 Hz sampling frequency, l is 80 times greater than s. So, the computation complexity is very high for ZTW$_{avg}$, ZTW$_{min}$, and ZTW$_{max}$ when compared to ZTW$_{uniform}$. The performance shows that ZTW$_{uniform}$ performed equally well with ZTW$_{avg}$. So, ZTW$_{uniform}$ is considered for future experiments.

5.2.3.2 Effect of different cepstral orders

Table 5.2 shows the performance in UAR% for i-vector based dialect classification system with ZTWCC and MFCC-ZTW features. The table shows that both ZTWCC and MFCC-ZTW performed well with lower cepstral orders (13 and 20). Based on this observation, the cepstral order is fixed to 20 for both ZTWCC and MFCC-ZTW.

Table 5.2: Performance (in UAR%) for i-vector dialect classification system with ZTW-based features (ZTWCC and MFCC-ZTW) with static cepstral coefficients for different cepstral orders.

	static cepstral coefficients					
Features/ #static coeff.	**13**	**20**	**30**	**40**	**50**	**60**
ZTWCC	72.75	**74.06**	73.68	72.84	69.17	66.14
MFCC-ZTW	**72.93**	71.06	68.71	66.75	67.86	64.33

5.2.3.3 Effect of different temporal contexts

Table 5.3 shows the performances in UAR% for i-vector based dialect classification system with different temporal contexts (static, static+Δ, static+Δ+ΔΔ, and static+SD coefficients) for both baseline (MFCC-STFT) and proposed (ZTWCC and MFCC-ZTW) features. From the table, overall, it can be observed that both proposed (ZTWCC and MFCC-ZTW) performed better than baseline (STFT-MFCC) features (weakly significant with $p < 0.1$ with p=0.07). Among ZTW based features, MFCC-ZTWs have shown better performance. Inclusion of delta coefficients in static+Δ, static+Δ+ΔΔ improved the performance of dialect classification system with ZTWCC and MFCC-ZTW. Static+SDCs of ZTWCC and MFCC-ZTW have improved over the static features. However, they didn't improve over static+Δ+ΔΔ despite their improvement in temporal context.

Table 5.3: Performance (in UAR%) for i-vector based dialect classification system with baseline (MFCC-STFT) and ZTW based features (ZTWCC and MFCC-ZTW) for all different temporal contexts such as static, static+Δ, static+Δ+ΔΔ, and static+SD coefficients.

Feat. type	**static**	**static+Δ**	**static+Δ+ΔΔ**	**static+SDCs**
MFCC-STFT	75.67	76.38	77.21	**77.98**
Rel. Imp.	-	0.94	2.04	3.05
ZTWCC	73.60	77.61	**78.23**	74.36
Rel. Imp.	-	5.45	6.29	1.03
MFCC-ZTW	71.73	78.32	**78.73**	75.85
Rel. Imp.	-	9.19	9.76	5.74

5.2.3.4 Comparison with confusion matrices

Figure 5.9 shows the confusion matrices for i-vector based dialect classification with baseline (MFCC-STFT) and proposed (ZTWCC and MFCC-ZTW) features. It can be observed that with ZTWCC and MFCC-ZTWCC, the performance of the UK class is improved. With ZTWCCs, the confusion for UK class with US is reduced, while with MFCC-ZTWCC, the confusion of AU class with US is reduced. This shows that for the differentiation of the UK class from the US class, non-linear mel scale analysis is better.

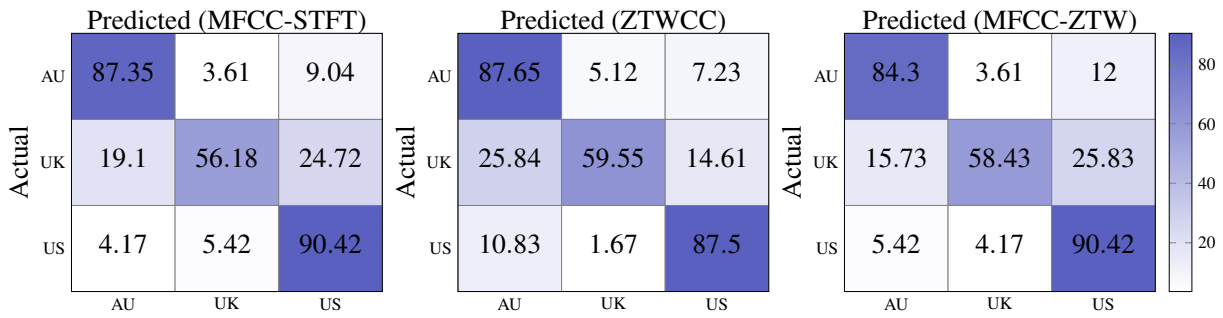
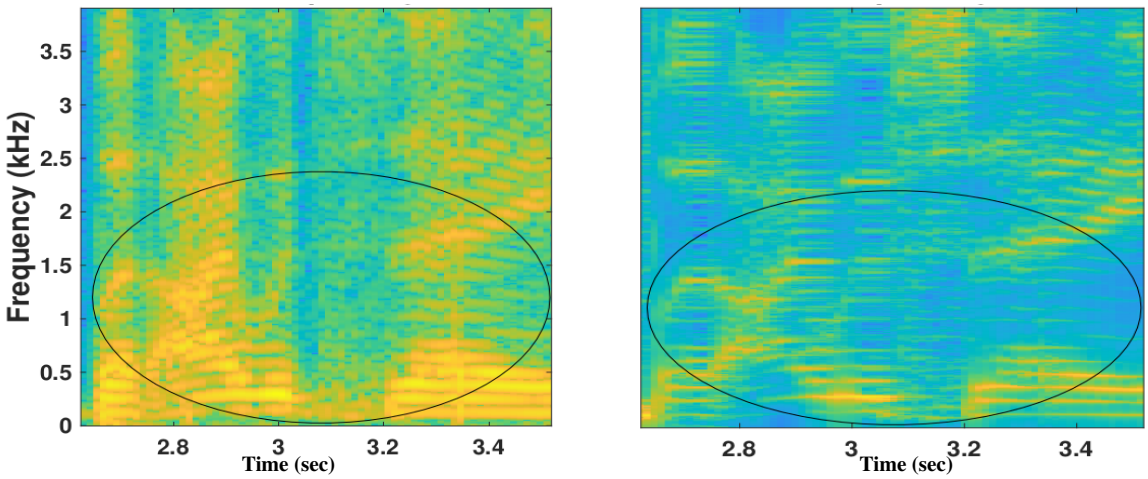


Figure 5.9: Confusion matrices for i-vector based dialect classification system with baseline MFCC-STFT (static+SDCs) features and proposed ZTW based features (static+Δ+ΔΔ of ZTWCC, MFCC-ZTW).

5.3 SFF based features

The spectrum computed by single frequency filtering (SFF) has been shown to give good spectral resolution to indicate harmonics and resonances [96] and good spectral resolution to model speech excitation features such as impulse-like events [97]. The SFF spectrum has also shown promising performance in determining burst-onset points related to voice-onset time (VOT) and glottal closure instances compared to the short-time Fourier transform (STFT) spectrum [97–99]. Previous studies in dialect classification have shown the significance of VOT for identification of accent [100].

Illustrations of spectrograms obtained with STFT and SFF methods are shown in Figures 5.10(a) and (b) respectively. From the figures, it can be clearly seen that SFF spectrogram (Figure 5.10(b)) highlights the harmonic structure (with sharper harmonics) compared to STFT spectrogram (Figure 5.10(a)), even though both of them show similar formant structure.



(a) **STFT Spectrogram** (b) **SFF Spectrogram**

Figure 5.10: Illustration of spectrograms obtained with (a) STFT and (b) SFF methods.

This section first describes the SFF method used for deriving high−resolution spectrum [106], and then gives a procedure to extract the proposed features from SFF spectrum.

5.3.1 SFF method

SFF [106] is a time-frequency analysis method that is used to compute an amplitude envelope of speech signal as a function of time at each of the selected frequency. In this method, the amplitude envelope at particular frequency is obtained by first frequency−shifting (i.e., modulating) the speech signal ($s[n]$), and then multiplying the $s[n]$ with an exponential function: $\hat{s}[n,k] = s[n]e^{j\hat{\omega}_k n}$, where $\hat{\omega}_k = \pi - \frac{2\pi f_k}{f_s}$, f_k is the desired frequency and f_s is the sampling frequency. The frequency-shifted signal is filtered using a single pole filter, whose transfer function is given by: $H(z) = \frac{1}{1+rz^{-1}}$. The pole of the filter is located on the negative real axis (at $z = -r$). In this study $r = 0.99$ is used, which is closer to the unit circle. The output of the filter is given by

$$y[n,k] = -ry[n-1,k] + \hat{s}[n,k]. \tag{5.6}$$

The amplitude envelope ($v[n,k]$) of $y[n,k]$ at frequency f_k is given by

$$v[n,k] = \sqrt{(y_r[n,k])^2 + (y_i[n,k])^2}, \tag{5.7}$$

where $y_r[n,k]$ is the real part and $y_i[n,k]$ is the imaginary part of $y[n,k]$. The amplitude envelopes can be computed for several frequencies at intervals of Δf by defining f_k as follows:

$$f_k = k\Delta f, \qquad k = 1,2,\ldots,K, \tag{5.8}$$

where $K = \frac{(f_s/2)}{\Delta f}$. In this study, the value of Δf is chosen such that 1024 frequency samples exist in between 0 to fs. From $v[n,k]$, the SFF magnitude spectrum (or SFF spectrum) can be obtained at each instant of time ('n') by considering all the amplitude envelope values at particular time instant. A schematic block diagram describing the steps involved in the computation of SFF spectrum is shown in Figure 5.11.

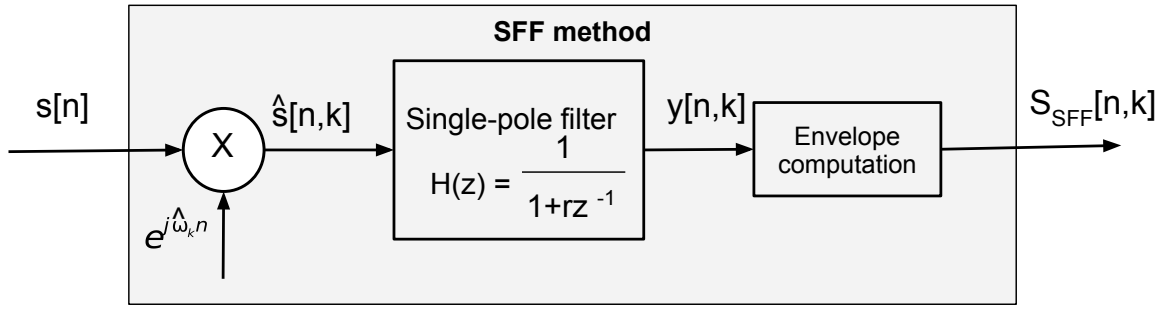


Figure 5.11: Schematic block diagram describing the steps involved in the computation of SFF spectrum.

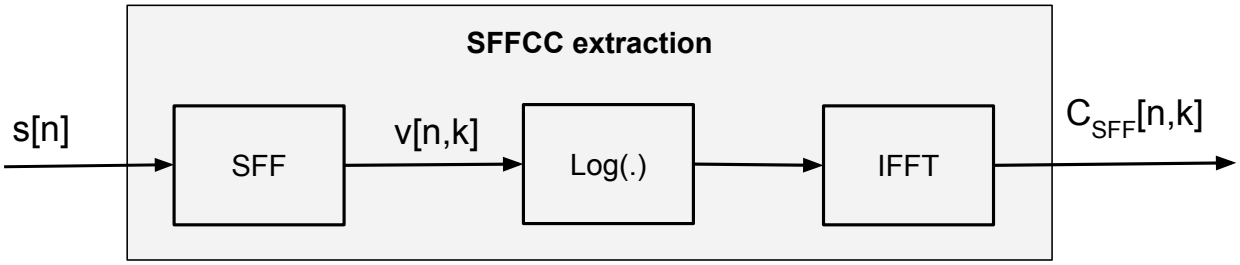


Figure 5.12: Schematic block diagram describing the steps involved in the computation of SFFCCs from SFF spectrum.

5.3.2 Extraction of feature representations from SFF method

This study proposes to derive two types of features from SFF spectrum. They are: (1) single frequency filtered cepstral coefficients (referred to as SFFCCs) and (2) mel frequency cepstral coefficients derived from SFF spectrum (referred to as MFCC−SFF). As per our knowledge, this is the first attempt to propose to use these feature representations for dialect classification. In principle, SFF spectrum can be obtained at each instant. Despite ZTW, approach the time complexity of uniform is same as average and based on the observations from [107], averaged SFF spectrum ($v[n,k]$) performed better [107]. So, the features are extracted from averaged spectrum at regular intervals of 12.5 msec.

5.3.2.1 Extraction of single frequency filtered cepstral coefficients (SFFCCs)

SFFCCs are computed from the cepstrum of SFF spectrum ($v[n,k]$), as follows [108]:

$$C_{SFF}[n,k] = \text{IFFT}(\log_{10}(v[n,k])). \tag{5.9}$$

From cepstrum $C_{SFF}[n,k]$, the first 80 coefficients are considered in this study. A schematic block diagram describing the steps involved in the extraction of SFFCCs is shown in Figure 5.12.

5.3.2.2 Extraction of MFCCs from the SFF spectrum (MFCC−SFFs)

A schematic block diagram describing the steps involved in the extraction of MFCC using the SFF spectrum is shown in Figure 5.13. The MFCC extraction consists of the mel filter−bank analysis on the SFF spectrum, followed by logarithm and discrete cosine transform (DCT) operations, and which can be expressed as follows:

$$MFCC_{SFF}[n,k] = DCT(\log(Mel(\mid v[n,k] \mid^2))), \tag{5.10}$$

where $MFCC_{SFF}[n,k]$ denotes the mel-cepstrum. The resulting cepstral coefficients are referred to as MFCC−SFF, and they represent compactly the spectral characteristics. From the mel-cepstrum, all 80 cepstral coefficients (including the zeroth coefficient) are considered.

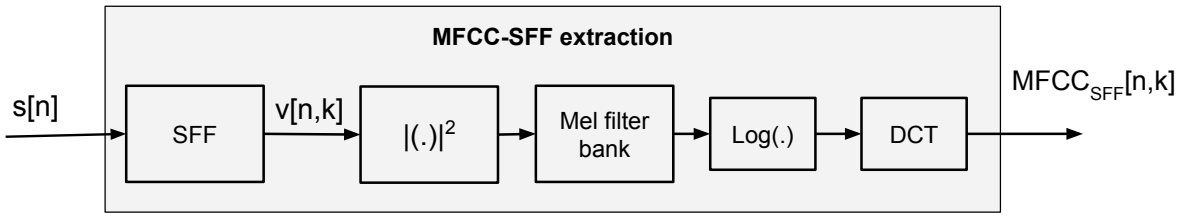


Figure 5.13: Schematic block diagram describing the steps involved in the computation of MFCC-SFF from SFF spectrum.

5.3.3 Results and discussion

First, this section provides experimentation results and discusses different cepstral orders for SFFCC and MFCC-SFF with i-vector based dialect classification. Then, it investigates different temporal contexts for SFF-based features. Finally, it compares the performances of SFFCC and MFCC-SFF to baseline MFCC-STFT using confusion matrices for dialect classification.

5.3.3.1 Different cepstral orders

This section investigated different cepstral orders ranging from 13 to 60. Table 5.4 shows the performances in UAR for i-vector based dialect classification system with different cepstral orders. Row 3 shows the performance of static SFFCCs, and Row 4 shows the performance of static MFCC-SFF features. From the table, it can be observed that both SFFCC and MFCC-SFF have shown better performance with cepstral order 20. So, future experiments were conducted with cepstral order 20.

5.3.3.2 Different temporal contexts

This subsection investigated different temporal contexts such as static+Δ, static+Δ+ΔΔ, and static+SD coefficients for dialect classification with SFF-based features. Table 5.5 shows the performances in UAR for the i-vector based dialect classification system for SFFCC (row 4) and MFCC-SFF (row 6) for different temporal contexts. The performance of the dialect classification system with baseline MFCC-STFT (row 2) was also included in the table for comparison. Relative Improvements (Rel. Imp.) with respect to static

Table 5.4: Performance (in UAR%) for i-vector based dialect classification system with SFF-based static cepstral coefficients varying the cepstral orders (from 13 to 60).

	static cepstral coefficients					
Features/ #static coeff.	**13**	**20**	**30**	**40**	**50**	**60**
SFFCC	70.46	**71.50**	68.89	**71.96**	70.76	68.39
MFCC-SFF	72.70	**73.26**	71.32	70.24	65.95	68.35

for different temporal contexts were also provided in the table (Rel. Imp. of STFT-MFCC, SFFCC, and MFCC-SFF in rows 3,5 and 7, respectively.).

From the table, it can be observed that both proposed SFFCC and MFCC-SFF performed better than MFCC-STFT (statistically strongly significant with $p < 0.005$). Among SFF-based features, MFCC-SFF performed better, showing the importance of mel-scale for dialect classification.

Table 5.5: Performance (in UAR%) for i-vector based dialect classification system with baseline (MFCC-STFT) and SFF based features (SFFCC and MFCC-SFF) for all different temporal contexts such as static, static+Δ, static+Δ+ΔΔ, and static+SD coefficients.

Feat. type	**static**	**static+Δ**	**static+Δ+ΔΔ**	**static+SDCs**
MFCC-STFT	75.67	76.38	77.21	**77.98**
Rel. Imp.	-	0.94	2.04	3.05
SFFCC	71.96	74.72	78.94	**79.10**
Rel. Imp.	-	3.84	9.70	9.92
MFCC-SFF	73.26	75.08	79.97	**81.25**
Rel. Imp.	-	2.48	9.16	10.91

With the inclusion of Δ coefficients with static (static+Δ), SFFCCs improved by 3.84% and MFCC-SFF improved by 2.48% UAR in relative. With the inclusion of Δ+ΔΔ with static coefficients, SFFCCs improved by 9.70% and MFCC-SFF improved by 9.16% UAR in relative. Including shifted delta coefficients with static (static+SD coefficients) improved the performance of SFFCC by 9.92% and MFCC-SFF by 10.91%. Overall, the performance has been enhanced with increased contextual information. Finally, with 160 coefficient of static+SDCs, MFCC-STFT gave a UAR of 77.98%, SFFCC gave a UAR of 79.10% for dialect

classification, and MFCC-SFF gave a UAR of 81.25% for dialect classification. SFF-based features SFFCCs and MFCC-SFFs have shown an improvement of 1.44% UAR and 4.19% UAR compared to baseline MFCC-STFT features for dialect classification.

5.3.3.3 Comparison to baseline and proposed with confusion matrices

This section reports the confusion matrices for i-vector based dialect classification system with baseline (MFCC-STFT) and proposed (SFFCC and MFCC-SFF) features. Figure 5.14 shows the confusion matrices. It can be observed that the performance of AU and US classes is above 80% for both baseline and proposed features. However, the performance of UK is low (below 60%). With SFF-based features, the performance of UK class has improved. With SFFCCs, the confusion for UK class with US is reduced, while with MFCC-SFFCC, the confusion of AU class with US is reduced. This observation is similar to ZTW and MFCC-ZTW features.

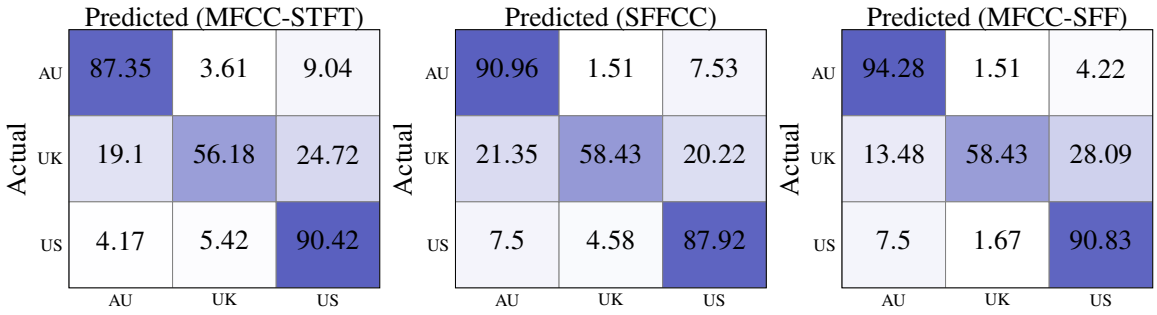


Figure 5.14: Confusion matrices for i-vector based dialect classification system with baseline MFCC-STFT (static+SDCs) features and proposed SFF based features (static+ SDCs of SFFCC, MFCC-SFF).

5.3.3.4 Fusion of SFF and ZTW features

To comprehend the complementary features between SFF and ZTW features, an experiment was conducted involving the fusion of SFFCC with ZTWCC, as well as MFCC-SFF and MFCC-ZTW. Table 5.6 presents the performance of the fusion of i-vectors derived from SFF and ZTW features.

Conducting an error analysis on the samples that were incorrectly predicted has led to the realization that there are numerous commonly misidentified samples between the SFF and ZTW-based features. Even though there is a slight improvement with fusion as shown in the table, the i-vectors extracted from SFF and ZTW-based features seem to not have any complimentary information that helps in dialect classification.

Table 5.6: Performance (in UAR%) of dialect classification with fusion of i-vectors derived from SFF with the i-vectors derived from ZTW based features.

Fusion	**Feat1**	**Feat2**	**UAR**
ZTWCC+SFFCC	78.23	79.10	80.27
MFCC-ZTW+MFCC-SFF	78.73	81.20	81.87

5.4 Summary and conclusions

The features (ZTW and SFF) that provide high temporal resolution without compromising spectral resolution are proposed in this chapter for dialect classification. From ZTW/SFF spectrum, along with cepstral coefficients, mel-frequency cepstral coefficients (MFCCs) were also extracted.

On comparison of ZTW-based features (ZTWCCs and MFCC-ZTWs) to best baseline features (MFCC-STFT), the proposed ZTW-based features performed better than baseline features for dialect classification. Among ZTW based features, MFCC-ZTW performed better with a UAR of 78.73%. This shows that the better temporal and spectral resolution of ZTW features is helpful in dialect classification.

On comparison of SFF-based features (SFFCCs and MFCC-SFFs) to baseline and proposed ZTW-based features, SFF based features outperformed. Among the SFF based features, MFCC-SFF performed better showing the importance of mel-scale for dialect classification with a UAR of 81.25%. The better performance of SFF based features shows that the better spectral features such as harmonics, resonances and time−domain features such as glottal closure instances and voice−onset time (VOT) are advantageous for dialect classification.

Chapter 6

Exploration of temporal dynamics of frequency domain linear prediction cepstral coefficients for dialect classification

Dialectal variations can be observed at the phonemic level, syllabic level, or sentence level. Phonemic level variations include the variations in the distribution of sounds and variations in articulatory trajectories within the same sound across dialects [109]. Syllabic level variations across dialects occur due to variations in stress patterns, intonation contour, duration, and articulatory trajectories based on the rules defined for respective dialect [109, 110]. Sentence level variations across dialects occur due to variations in sentence-level intonation and higher-level linguistic factors such as usage of words, i.e., vocabulary [111]. From the above discussion, it is evident that the dialect discriminant information can be found not only by observing a single sound unit (phonemic/syllabic), but also temporal dynamics across the sound units.

Conventional short-term spectral features such as mel frequency cepstral coefficients (MFCCs) are derived by windowing the signal with a window of length 10-30 msec and incorporate weak temporal context using delta coefficients (Δ, and $\Delta\Delta$), and shifted delta coefficients (SDCs) [22, 80]. These windowed representations may fail to represent the instantaneous burst representations of stops and fricatives and also may fail to represent temporal dynamics across windows [112, 113].

Representation for temporal dynamics of the speech signal can be obtained at the acoustic level or at the phonetic level. Acoustic-level temporal dynamics can be represented by segmenting the speech signal into syllables either manually or automatically. In [24], the speech signal is segmented into pseudo-syllables, and the acoustic variations such as pitch, rhythm, and duration are investigated for dialect classification. In [114], supra-segmental prosodic variations obtained from pseudo-syllables are modeled using n-gram language model. To take the advantage of temporal context, two models (stochastic trajectory model (STM) and parametric trajectory model (PTM)) are investigated on segmental cepstral coefficients in [35]. Temporal dynamics can also be modelled using higher linguistic features such as phones [25–28, 115–117]. The methods in this approach involves a phone recognizer and modelling techniques such as phone recognition followed by language model (PRLM) and parallel-PRLM (PPRLM) [25–28]. These approaches require an

external phone recognizer and often the dialect identification accuracy depends on the performance of phone recognizer. To overcome this, we investigate the effectiveness of acoustic features that captures the longer temporal context.

In the present study, the effectiveness of frequency domain linear prediction cepstral coefficients (FDLPCCs) which has the ability to capture the longer temporal context are investigated for dialect classification. Traditional linear prediction, i.e., time-domain linear prediction (TDLP) analysis estimates the spectral peaks by computing auto-correlation of a signal. By duality principle, frequency domain linear prediction (FDLP) estimates temporal peaks by computing auto-correlation of discrete cosine transform (DCT) sequence [112, 118–121]. Unlike conventional short-term spectral feature extraction methods, the sub-band FDLP envelope captures extended temporal context as the estimated temporal peaks are the resultant of long-timescale summarization [112]. We hypothesize that the long temporal nature of FDLP spectrum may be advantageous in discriminating dialects.

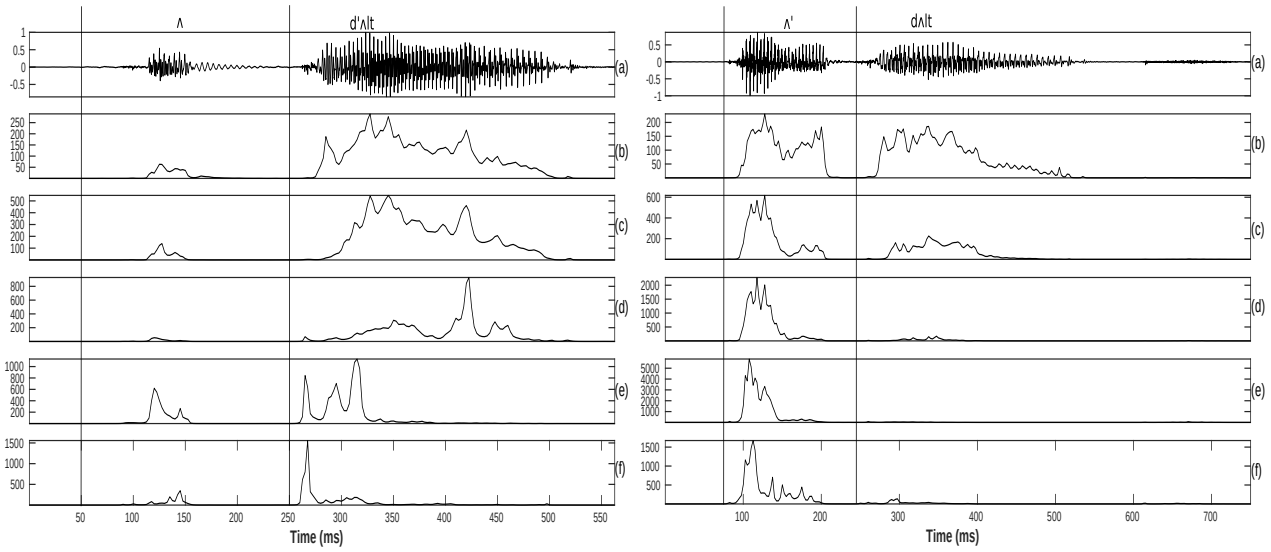
Figure 6.1 illustrates the temporal variations in terms of amplitude envelopes across five sub-bands (i.e., Figure 6.1 (b)-(f)) for the word 'adult' spoken by an American speaker (shown in Figure 6.1 (i)) and by a British speaker (shown in Figure 6.1 (ii)). The speech signals are shown in subplots (a) in Figure 6.1. "Adult" is one of the loan words from French. Such words were adapted differently between US and UK dialects. Americans (US) emphasize on second syllable, while British (UK) emphasize on first syllable [7]. The stress assignment for the word "adult" is [ʌdʌ'lt] for American speaker and [ʌ'dʌlt] for British speaker. The phones are segmented and stress is represented by '. From the figure, it can be clearly observed that the temporal variations in stress patterns between American and British speakers are different. American speaker stressed on the second syllable (see Figure 6.1 (i) in time interval of 250 to 500 msec) while the British speaker stressed on the first syllable (see Figure 6.1 (ii) in time interval of 100 to 200 msec) of a bi-syllabic word. Inspired by this observation, FDLP based cepstral coefficients are investigated for dialect classification in this study.

The deep neural network (DNN) architectures with convolution neural network (CNN) and time delay neural network (TDNN) models were investigated in the previous studies [9, 32, 45, 84, 122–126] which could capture long temporal context. They are also compared to previous studies that used UT-Podcast using DNN architectures [84].

The contributions of this study are as follows:

- Application of FDLPCCs for dialect classification based on the hypothesis that FDLP captures the longer temporal dynamics.

- Analysis of different temporal context representations such as delta and double delta $(\Delta + \Delta\Delta)$, and shifted delta cepstra (SDC) coefficients for baseline and proposed features.



(i) American speaker (ii) British speaker

Figure 6.1: Illustration of sub-band temporal envelopes estimated using FDLP for the word 'adult' spoken (i) by an American speaker and (ii) by a British speaker.

6.1 FDLPCCs feature extraction

Frequency domain linear prediction (FDLP) is an efficient method for auto regressive (AR) modelling of temporal envelopes of speech signal [112, 118–121]. The AR model approximates the power spectrum of the speech signal in time domain linear prediction (TDLP), whereas in FDLP, an all pole model is fitted to the Hilbert envelope (squared magnitude of the analytic signal). As the estimated temporal peaks are the resultant of longer time signal, they capture finer details of the linguistic units. We hypothesize that the long temporal nature of FDLP spectrum may be advantageous in discriminating dialects. The extraction of frequency domain linear prediction cepstral coefficients (FDLPCCs) from speech signal involves two stages as shown in Figure 6.2. The first stage (first seven blocks in the figure) involves the estimation of sub-band temporal envelopes and the second stage (next three blocks in the figure) involves the extraction of cepstral coefficients from sub-band FDLP envelopes. The steps involved in estimation of sub-band FDLP envelopes are described in Section 6.1.1, and the extraction of cepstral coefficients (i.e., FDLPCCs) from FDLP envelopes are described in Section 6.1.2.

6.1.1 FDLP method

This section describes the steps involved in the estimation of sub-band FDLP envelopes from speech signal [119]. They are:

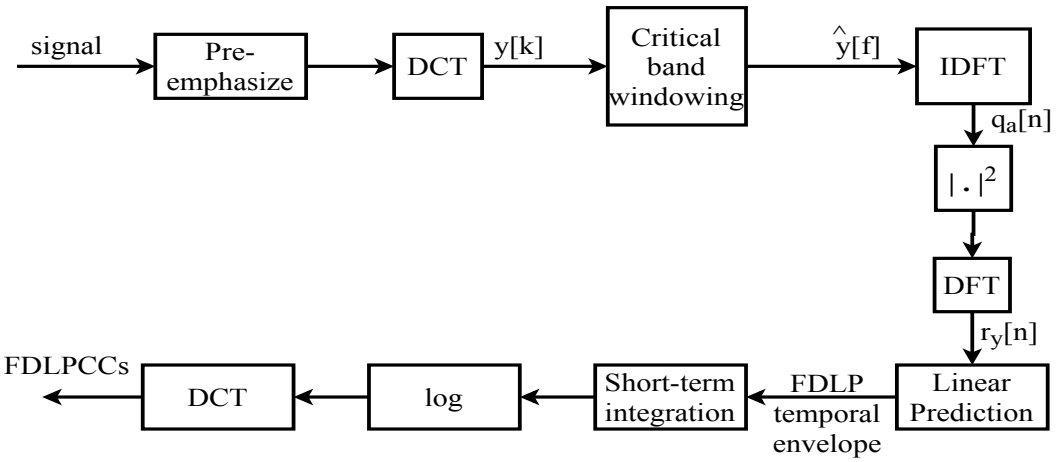


Figure 6.2: Block diagram describing the steps involved in extraction of FDLPCCs.

- Speech signal $s[n]$ is pre-emphasized to remove the low frequency variations caused due to recordings, and to emphasize high frequency components.

$$x[n] = s[n] - \alpha s[n-1] \tag{6.1}$$

- DCT full-band sequence is computed by applying DCT over the pre-emphasized signal ($x[n]$) for every second. Unlike short-time segmental feature extraction methods, spectral transformation is done over a long temporal signal.

$$y[k] = a[k] \sum_{n=0}^{N-1} x[n] cos\left(\frac{(2n+1)\pi k}{2N}\right), \tag{6.2}$$

where $k = 0, 1, 2 \ldots N-1$ and

$$a[k] = \begin{cases} \frac{1}{\sqrt{N}} & k = 0 \\ \sqrt{\frac{2}{N}} & k = 1, 2, \ldots, N-1 \end{cases}$$

- Sub-band DCT components are derived by windowing the full-band DCT sequence. The sub-band DCT sequence for a band f (critical band windowing) is represented by $\hat{y}[f]$.
- Analogous to TDLP, applying DFT over the squared magnitude of analytic signal gives auto-correlation of spectral coefficients. The inverse DFT (IDFT) of zero-padded DCT sequence is called even symmetric discrete time analytic signal. The analytic signal derived from each sub-band DCT component is given by:

$$q_a[n] = IDFT(\hat{y}[f]) \tag{6.3}$$

Autocorrelation coefficients for each sub-band spectrum $\hat{y}[f]$ is derived by applying DFT over each sub-band analytic signal, as given by:

$$r_y[\tau] = DFT(|q_a[n]|^2) \tag{6.4}$$

- Similar to TDLP, these autocorrelations are used to obtain linear prediction coefficients that are smoothed approximation of sub-band Hilbert envelopes. The LP order (or pole order) to estimate LPCs modulate the efficient representation of sounds. The approximation of sub-band Hilbert envelopes estimated using LPCs is referred as sub-band FDLP envelope in this study. The sub-band FDLP envelope captures extended temporal context as the estimated temporal peaks are the resultant of long-timescale summarization.

6.1.2 Extraction of FDLPCCs

- Energies in a set of sub-band FDLP envelopes are integrated in a long-term analysis window to obtain FDLP short-term frames. To be analogous to short-time segmental feature extraction methods, the window length and window shift are similar to conventional methods.
- DCT is applied over the logarithm of integrated FDLP energies across sub-bands within a frame to obtain FDLPCCs for each frame.

6.1.3 Parameters used for FDLPCCs extraction

In this study, the entire signal is considered to obtain the full-band DCT sequence, and then the DCT sequence is multiplied with mel-band Gaussian windows. Typically, the number of mel-band Gaussian windows are given by:

$$n_{mel-bands} = \lceil \mathrm{F}_{\mathrm{hz2mel}}(\frac{fs}{2}) \rceil, \tag{6.5}$$

where fs is the sampling frequency in Hertz (Hz) and $\mathrm{F}_{\mathrm{hz2mel}}$ is a function that converts Hz to mel using Slaney's auditory toolbox [127] which will result in 37. However, a different number of mel-bands such as 13, 37, 80, 128 and 160 were investigated and it was observed that 37 and 80 mel-bands gave better performance compared to others. In all the experiments of the study, 37 mel-bands are used.

Autocorrelation formulation of linear prediction is used to estimate temporal poles for each sub-band FDLP envelope. The number of temporal poles is set to 160, similar to previous studies [120]. The gain normalized sub-band FDLP temporal envelopes are integrated along the time axis within a window of 25 msec, and half of it is used as window shift. Static FDLPCCs are obtained by applying DCT over the logarithm of integrated FDLP energies across sub-bands within a frame. We investigated the effect of a number of static cepstral coefficients (varying from 13 to 60) on the performance of dialect classification. From static coefficients, Δ+$\Delta\Delta$ and SDC coefficients [80] are also derived, which are also investigated to see their effectiveness on dialect classification [1].

[1] https://github.com/iiscleap/FeatureExtractionUsingFDLP

6.2 Results and discussion

This section investigates i-vector representations derived from baseline (MFCC-STFT) and proposed (FDLPCC) features for dialect classification. To find the best configurations of FDLPCC features, i-vector representations are derived from different static cepstral orders varying from 13 to 60 (13, 20, 30, 40, 50, and 60). Further, both baseline and proposed features are investigated with different temporal contexts (i.e., static+Δ, static+Δ+ΔΔ and static+SDCs). To summarize all the proposed features of the thesis, FDLPCC features are compared to the baseline and proposed features from previous chapters. The existence of complementary information is investigated in Section 6.2.3 by fusing at utterance-level (U-level) of MFCC-STFT, ZTW/SFF-based features with FDLPCCs for dialect classification. Finally, the performance of all the proposed features is compared to previous studies.

6.2.1 Effect of cepstral order and temporal context

Table 6.1 shows the performances for the static FDLP cepstral coefficients, by varying the number of static cepstral coefficients from 13 to 60 (13, 20, 30, 40, 50, and 60). From the table, it can be observed that FDLPCCs performed better for 20−dimensional static coefficients.

Table 6.1: Performances (in UAR%) for i-vector based dialect classification for FDLPCC features with static cepstral coefficients, by varying cepstral coefficients dimension from 13 to 60 (13, 20, 30, 40, 50, and 60).

	static cepstral coefficients					
Features/ #static coeff.	**13**	**20**	**30**	**40**	**50**	**60**
FDLPCC	71.8	**77.3**	76.2	68.1	67.8	66.2

Table 6.2 shows the performances for the baseline and the proposed features with static, static+Δ, static+Δ+ΔΔ, and static+SD coefficients. From the table, it can be observed that the inclusion of Δ coefficients improved the performance of FDLPCC by 4.64% in relative. Including Δ+Δ coefficients with static improved the performance of FDLPCC by 5.22% UAR in relative. Including shifted delta coefficients with static improved the performance by 2.21% UAR in relative. This shows that temporal context is required for dialect classification with FDLPCC features. However, due to its inbuilt temporal summarization, it doesn't improve with a longer temporal context provided by SDC when compared to ΔΔ coefficients.

Table 6.2: Performance (in UAR%) for i-vector based dialect classification system with baseline (MFCC-STFT) and proposed (FDLPCC) features for all different temporal contexts such as static, static+Δ, static+Δ+ΔΔ, and static+SD coefficients.

Feat. type	**static**	**static+Δ**	**static+Δ+ΔΔ**	**static+SDCs**
MFCC-STFT	75.67	76.38	77.21	**77.98**
Rel. Imp.	-	0.94	2.04	3.05
FDLPCC	77.33	80.92	**81.37**	79.04
Rel. Imp.	-	4.64	5.22	2.21

6.2.2 Comparison to other proposed features

Table 6.3 shows the performance (in UAR% and class-wise accuracies) of i-vector based dialect classification system with the baseline (MFCC-STFT) and proposed features (ZTWCC, MFCC-ZTW, SFFCC, MFCC-SFF, and FDLPCC). Dialectal variations can be observed either at frame level or across frames. So, the ZTW and SFF based features that provide higher spectral and temporal resolution and FDLP based features that provide longer temporal summarization are investigated. From the results, it can be observed that both higher spectral and temporal resolution and longer temporal summarization are important for dialect classification based on the performance of the proposed features. The

From class-wise accuracies of the table, it can also be observed that proposed features performed well in discriminating with minority class (UK).

Table 6.3: Performances (in UAR% and class-wise accuracies) for the baseline and proposed (ZTWCC, MFCC-ZTW, SFFCC, MFCC-SFF, and FDLPCC) features with the best configurations (i-vector approach).

Features/Class	**UAR**	**AU**	**UK**	**US**
MFCC-STFT (static + SDC)	**77.98**	87.35	56.18	90.42
ZTWCC (static +Δ+ΔΔ)	**78.23**	87.65	59.55	87.50
MFCC-ZTW (static +Δ+ΔΔ)	**78.73**	87.35	58.43	90.42
SFFCC (static+SDC)	**79.10**	90.96	58.43	87.92
MFCC-SFF (static+SDC)	**81.20**	**94.28**	58.43	**90.83**
FDLPCC (static +Δ+ΔΔ)	**81.37**	**86.14**	**66.29**	**91.67**

Among the proposed features, MFCC-SFF and FDLPCC outperformed all the other features. Similar to the baseline and previously proposed features, the performance of AU and US classes is above 85%. Considering the accuracy of the UK class, FDLPCCs outperformed all the other features with an accuracy of 66.29%.

Figure 6.3 shows the confusion matrices of baseline (MFCC-STFT) and proposed (ZTW/SFF-CC, MFCC-ZTW/SFF, and FDLPCC) for dialect classification. FDLPCCs reduced the confusion of the UK class with the AU class when compared to baseline features while the confusion of the UK with the US remained the same. FDLPCCs exhibited similar behaviour as MFCC-ZTW/SFF which reduced the confusion of the UK with AU class.

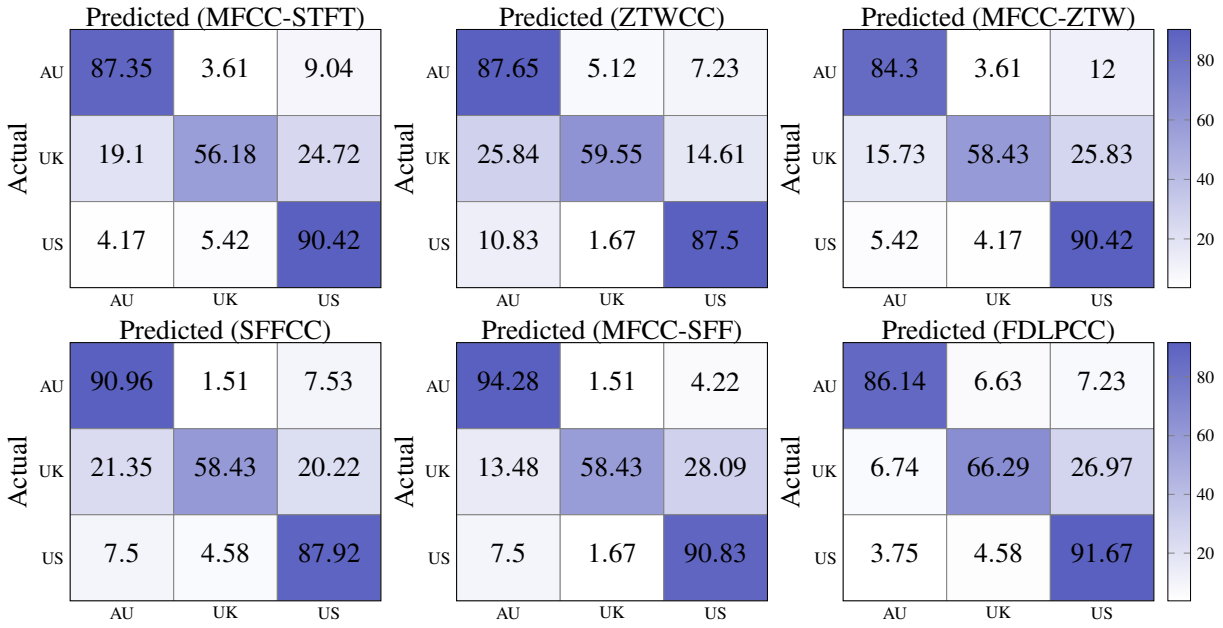


Figure 6.3: Confusion matrices for i-vector based dialect classification system with baseline (MFCC-STFT) and proposed (ZTWCC, MFCC-ZTW, SFFCC, MFCC-SFF, and FDLPCC) features.

The performances of MFCC-SFF and MFCC-ZTW have improved when compared to MFCC-STFT for UK and AU accents in the figure. The features derived from SFF and ZTW are hypothesized to represent the intrinsic dynamic features such as trills, nasals, approximants, and fricatives [97–99, 101–103]. Let us understand how SFF and ZTW could have helped in dialect classification with an example. UK and AU dialects are non-rhotic as compared to the US dialect where the strong pronunciation of /r/ is observed. The representation of the transient behaviour /r/ by SFF and ZTW features might have helped to differentiate UK and AU dialects from US dialect. This could have been the reason for the best behaviour of AU and UK with MFCC-ZTW and MFCC-SFF.

From the analysis of acoustic correlates in [128] with pitch, it is observed that British speakers possess the steepest rate of initial pitch rise among the three accents about 44.1% and 22.7% steeper than Australian and American respectively. This coincides with the fact that they have the largest frequency range of three. Similarly, British speakers possess the sharpest final fall rate in pitch compared to Australian and American speakers. Results illustrate that British speakers tend to have a steeper pitch rise and fall rates than American speakers. Furthermore, American speakers tend to have a lower pitch in the final words of sentences compared to British speakers. This differentiating feature of the UK dialect which could be represented better by FDLP, could be the reason for standing out performance of UK dialect in FDLPCC when compared to MFCC-STFT.

6.2.3 Existence of complementary information

To know the existence of complementary information between features that provide long temporal summarization and spectral features, experiments are carried out by fusing at utterance level (U-level) i.e., fusion of i-vectors. In the U-level fusion of the i-vector approach, 100-dimensional i-vectors are extracted from each of the baselines and proposed features, resulting in 200-dimensional i-vectors. Table 6.4 shows the performances (in UAR%) of fusion experiments (columns 5 and 6) along with individual feature performances (columns 3 and 4) for dialect classification.

Further, to investigate complementary information among features that provide high resolution (SFFCC, MFCC-SFF, ZTWCC, MFCC-ZTW), along with baseline (MFCC-STFT) with features that provide longer temporal summarization (FDLPCC), fusion experiments are conducted and reported in Table 6.4. From the results, it is observed that i-vectors derived from FDLPCCs, when combined with i-vectors derived from all spectral features, showed an improvement in performance for dialect classification. Fusion of FDLPCCs with MFCC-STFT (row 3) gave a relative improvement of 6.69% and 2.25% (in UAR relative) when compared to individual performances, MFCC-STFT and FDLPCCs in order. While the fusion of ZTWCC and MFCC-ZTWCC (rows 6 and 7) have a relative improvement of 1.93% and 2.72% (in UAR relative) when compared to best-performing FDLPCCs, respectively. Fusion of FDLPCCs with SFFCC and MFCC-SFF (rows 4 and 5) gave a relative improvement of 3.97% and 3.23% (in UAR relative) when compared to FDLPCCs, respectively. Among all the combinations, FDLPCCs, when combined with SFF-based features, have shown more significant improvement in the performance of dialect classification.

6.2.4 Comparison of current studies with previous studies

This section compares the results obtained in the current study (i-vectors derived baseline and proposed features with SVM) with the previous studies [72, 84]. In [72], both text based and audio based approaches were investigated. In text based approach, term-frequency and inverse document frequency (TF-IDF) were

Table 6.4: Performance (in UAR%) of dialect classification with fusion of i-vectors derived from FDLPCC features with the i-vectors derived from MFCC-STFT, SFF/ZTW based features.

Approach	**Fusion of feats** (Feat1 + Feat2)	**UAR**		**Fusion**
		Feat1	**Feat2**	
i-vectors	MFCC-STFT+FDLPCC	77.98	81.37	83.20
	SFFCC+FDLPCC	79.10		**84.60**
	MFCC-SFF+FDLPCC	81.20		84.00
	ZTWCC+FDLPCC	78.23		82.94
	MFCC-ZTW+FDLPCC	78.73		83.58

exploited. TF-IDF measures the originality of a word in a document. In audio based approach, GMM super-vectors and i-vectors were used with the SVM classifier. A fusion of both text and audio approaches is also investigated. In [84], DNN classifiers such as feed-forward neural network (FFNN), five-layer convolution neural network (CNN), AlexNet, VGG-11, and ResNet-18 trained with STFT-spectrogram are investigated. In this study, the corpus is modified by segmenting the utterances of the UK dialect to handle imbalanced classes. FFNN is a DNN classifier with three fully connected layers. A five-layer CNN is a DNN classifier with five convolution layers for segmental-level processing and fully connected layers for utterance-level processing. The other DNN classifiers, AlexNet [129], VGG-11 [130], and ResNet-18 [131] are typical deep architectures with a varied number of convolution layers. FreqCNN is proposed in [84], and its architecture comprises attention based convolution blocks along with basic convolution blocks.

Table 6.5 shows the performance of dialect classification (in UAR% and class-wise accuracies) for previous and current studies. From the first set of previous studies (rows 3-6) shown in Table 6.5, it can be observed that audio based approaches performed better than text based approaches. Within the audio based approaches, the i-vector approach performed better than the GMM approach. The fusion of audio (i-vectors) and text based systems has shown an improvement in performance by 2.4% relative (in UAR) than the i-vector system alone.

From the second set of previous studies (rows 8-13) shown in Table 6.5, it can be observed that their proposed complex FreqCNN architecture performed better than all the simple neural network architectures.

On comparison of current thesis studies to previous ones, it can be observed that current studies outperformed all the text and audio-based conventional approaches. The performance of conventional i-vector systems with SFF-based and FDLPCC is better than a complex neural network (FreqCNN).

Table 6.5: Comparison of current i-vector based dialect classification (with baseline and proposed features) with previous dialect classification models over UT-Podcast corpus (in UAR% and class-wise accuracies).

Arch. type	**UAR**	AU	UK	US
Text and audio based approaches from previous studies [72]				
Audio System (GMM)	**60.3**	85.5	32.6	62.9
Audio System (i-vector)	**74.5**	78.0	61.8	83.8
Text System (TF-IDF logistic regression)	**58.7**	83.1	32.6	60.4
Audio-Text system (Fusion)	**76.3**	86.1	60.7	82.1
DNN classifier from previous studies [84]				
FFNN	**61.4**	70.8	50.6	62.9
Five-layer CNN	**62.8**	64.8	41.6	82.0
AlexNet	**64.9**	58.4	64.0	74.2
VGG-11	**54.4**	55.7	48.3	59.2
ResNet-18	**61.7**	69.3	38.2	77.5
FreqCNN	**79.32**	88.55	71.91	77.50
i-vector system with SVM (current study)				
MFCC-STFT (static + SDC)	**77.98**	87.35	56.18	90.42
ZTWCC (static +Δ+ΔΔ)	**78.23**	87.65	59.55	87.50
MFCC-ZTW (static +Δ+ΔΔ)	**78.73**	87.35	58.43	90.42
SFFCC (static+SDC)	**79.10**	90.96	58.43	87.92
MFCC-SFF (static+SDC)	**81.20**	**94.28**	58.43	**90.83**
FDLPCC (static +Δ+ΔΔ)	**81.37**	**86.14**	**66.29**	**91.67**

However, it can be observed that FreqCNN has shown better performance in the classification of the UK class.

6.3 Summary and conclusions

Dialectal variations can be observed either at frame level or across frames. So, the ZTW and SFF-based features that provide higher spectral resolution without compromising temporal resolution and FDLP-based features that provide longer temporal summarization are investigated. From the results, it can be observed that both a higher degree of spectral resolution and longer temporal summarization are important for dialect classification based on the performance of proposed features when compared to the conventional STFT-MFCC features. Overall, SFF-based and FDLPCC features performed better for dialect classification.

On investigation of different temporal contexts, it is observed that the SFF features that provide good temporal resolution with higher temporal context improved performance. However, FDLPCCs that in-built have long temporal summarization didn't show greater improvement with shifted delta coefficients. From the fusion of FDLPCC with ZTW/SFF features, it can be observed that FDLPCCs and SFF-based features have more complementary information. With the proposed features, it can be observed that the performance of minority class UK is improved when compared to baseline features. We propose to study deep neural networks for dialect classification with proposed features.

Chapter 7

Deep neural architectures for dialect classification

Modern end-to-end deep neural classifiers can handle both compression and classification [125,132,133]. The compressed latent representations learnt from these networks retain dialect discriminative information and temporal dependencies across the frames. Deep neural classifiers were mainly investigated with convolution neural networks (CNNs) and recurrent neural networks (RNNs) for dialect classification [84, 125, 132–134]. From studies by [82, 125], it was found that compared to traditional statistical methods (i-vectors+SVM), the end-to-end CNN architectures performed better by 10% absolute in accuracy for Arabic English dialects.

Until now, this thesis proposed to leverage the signal processing approaches that provide good temporal and spectral resolution (single frequency filtering (SFF) and (zero-time windowing (ZTW) based) and long temporal summarization (frequency domain linear prediction (FDLP) based) for dialect classification. It is observed that these proposed features performed better than baseline features with an i-vector based classification system. The i-vectors are extracted in an unsupervised manner and result from a linear transformation of Gaussian mean vectors. The embeddings derived from DNNs are extracted in a supervised manner, and the resulting embeddings are the result of non-linear transformations. This non-linear transformation changes the correlation between variables, and the supervised learning leads to retaining only dialect discriminant information. To understand the advantages of proposed features with DNNs, this chapter investigates different DNN architectures that provide different temporal contexts.

The i-vector system uses different contextual processing approaches to improve classification. From experimentation, it is observed that the temporal context was helpful in the classification of dialects. The delta representations computed in the i-vector system result from linear transformation. The embeddings derived from DNNs are the non-linear weighted transformation of the neighbouring frames, and these weights are learnt by improving the performance for dialect classification (i.e., supervised approach).

Based on the advantages and effectiveness of deep neural networks for classification, different architectures of deep neural networks are investigated in this chapter. The DNN classifiers require a larger amount of data for training. The literature showed that data augmentation improved the performance by

5.5% absolute accuracy [82,125]. To overcome this, different data augmentation approaches are investigated as part of this thesis. Different weight initialization of neural networks can lead to unstable performances. To mitigate this, neural networks are trained multiple times and tested against each trained model, and then the statistics of the performance are reported.

Even though RNNs were used for classification tasks in speech as they capture long temporal context, they also require $O(n)$ sequential operations for each unit. In contrast, CNNs require $O(1)$ sequential operations. Lower order sequential operations for CNN lead to parallelization of computations in CNNs. In contrast, higher order sequential processing will lead to higher computation time for RNNs. Time-delay neural networks (TDNNs) [135] that provide higher temporal context than CNNs with similar computation complexity and temporal convolution neural networks (TCNNs) [136] that provide longer temporal context from only past with similar computation complexity are explored for dialect classification. Significant architectural changes were made to TDNN to obtain emphasized channel attention, propagation, and aggregation in TDNN (ECAPA-TDNN), which was shown to improve the performance of speaker verification system [137] and language identification [138] is investigated for dialect classification.

DNNs are investigated as part of the second step of the thesis plan to improve dialect classification (as in Figure 1.3). Figure 7.1 shows the schematic block diagram of the proposed dialect classification system with deep neural networks (DNNs). The proposed system consists of mainly three stages; (1) feature extraction, where feature representations from ZTW, SFF, and FDLP-based methods are derived, (2) embeddings extraction, and (3) classification of dialects. In DNNs, both embeddings extraction and classification are performed by DNN architectures such as CNN, TCNN, TDNN, and ECAPA-TDNN. Deep neural classifiers are trained with frame-level features from an entire utterance. They are trained to extract better embeddings and classify dialects better.

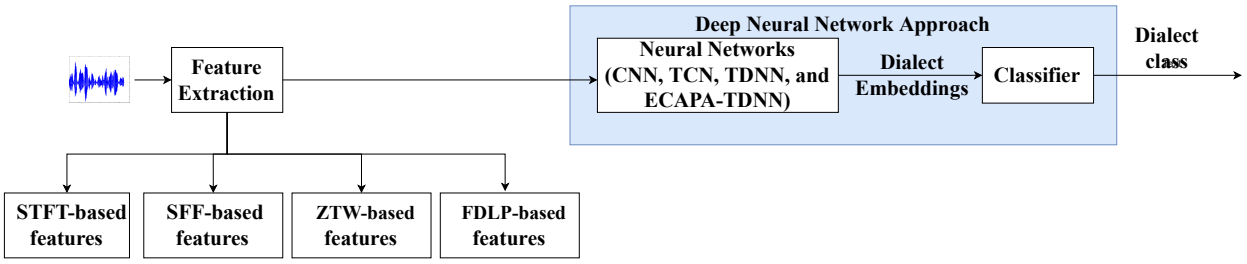


Figure 7.1: A schematic block diagram of the proposed deep neural network approach for dialect classification.

7.1 Convolution neural networks

CNNs are the most widely used deep neural architectures in speech [139], text [140], and image processing [141]. CNNs were investigated previously for dialect classification with 1D convolutions [125] and 2D convolutions [84]. A convolution neural network is usually formed by convolution layers (Conv), max-pooling, and fully connected (FC) feed-forward layers. The Conv layers of CNN extract the translation invariant and localized temporal features by striding over windows. The average pooling layer compresses the segmental-level information derived from the convolution layer to utterance-level information. FC layers are trained to classify the dialects. CNN with 1D convolution layers is investigated for dialect classification.

7.1.1 Architecture

The initial CNN architecture was similar to the one presented in [125]. Subsequently, modifications were made to enhance performance, particularly concerning MFCC-STFT features. The primary alteration involved introducing max pool layers and adjusting stride values. This study utilized a CNN with four convolution layers and three fully connected layers. Table 7.1 illustrates the architecture of the CNN classifier, with columns representing the layers and configurations defined along rows. Convolution layer configurations include the number of filters (filters), filter size, and stride, while max-pool layers are defined by kernel size and stride. Fully connected layers are defined by input and output dimensions.

Table 7.1: End-to-end CNN architecture for dialect classification. Conv represents the convolution layer, and FC represents a fully connected layer.

Layers:	**Conv1**	**Conv2**	**Max pool**	**Conv3**	**Conv4**	**L2 pool**	**FC1**	**FC2**	**FC3**
# filters/output dim.	500	500	-	3000	3000	3000	1500	600	3
Kernel size	5	3	10	5	3	-	-	-	-
Stride	1	1	10	1	1	-	-	-	-

Figure 7.2 is a block diagram of CNN architecture showing all the layers. Each layer is defined by [outputsize]-[kernalsize]K-[stride]S. Convolution and max-pooling layers are segmental layers, and the layers after L2 pool processes on utterance-level representations. Average pooling is done after convolution layers to convert frame-level features to utterance-level representations. The fully connected layers (FC1, FC2, and FC3) learn to classify dialects from the utterance-level representations. Rectified linear unit (ReLU) activation is commonly applied in all the layers.

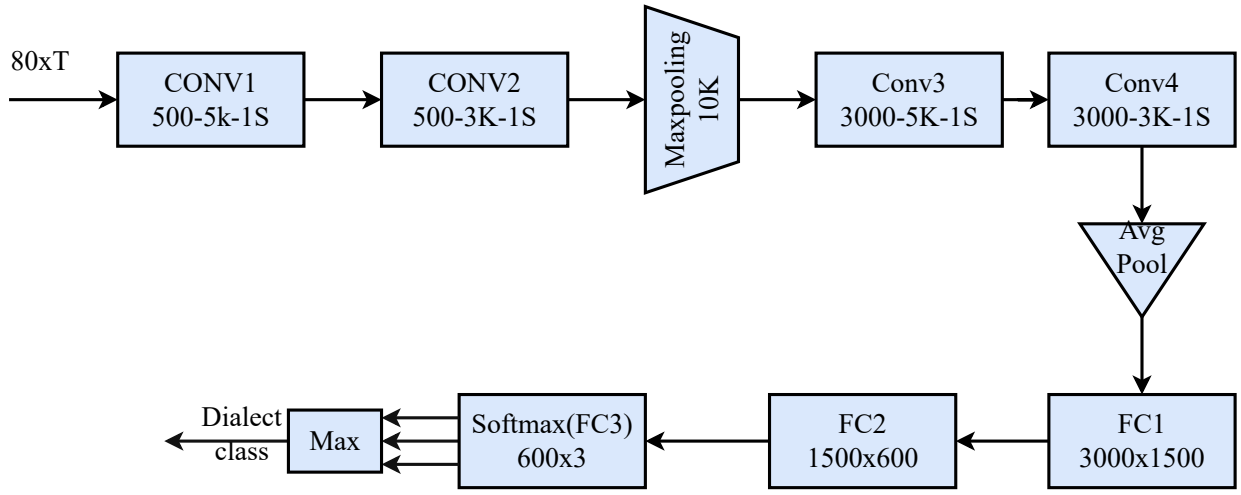


Figure 7.2: A schematic block diagram showing architecture of convolution neural network.

7.1.2 Experimental Protocol

CNN is investigated by training with both the baseline and proposed features. The number of training epochs is decided approximately based on the loss convergence and overfitting. CNN is trained for around 70 epochs with cross-entropy loss. A gradient descent optimizer with a learning rate of 0.001 is used to train the model. To mitigate the side-effect of the neural network weights initialization, networks are trained multiple times (six times for all the experiments) and tested against each trained model. The performance is averaged across all models, and the mean and standard deviation of UAR% are reported for all the experiments.

For hyperparameter tuning, k-fold validation is done using only the train data. With the best parameters settings, the new model is trained again with all the train data.

7.1.3 Results and discussion

The performance of the dialect classification system in the mean and standard deviation of UAR% and mean of class-wise accuracies from six trials with CNN classifier are reported in Table 7.2. The third column of the table reports the mean and standard deviations of UAR%, and the fourth to sixth columns report class-wise accuracies. The performances of baseline (MFCC-STFT) and proposed features (ZTWCC, MFCC-ZTW, SFFCC, MFCC-SFF, and FDLPCC) with the CNN classifier are reported in the table. Overall, it can be observed from the table that all the proposed features performed better than baseline features with CNN.

However, the performances are very poor, especially the performance of the UK class is very low with baseline features (11.99% accuracy for the UK). This is due to unequal strengths of the classes in the corpus

during training, which led to biased predictions towards the majority classes (AU and US). Section 7.1.4 investigated different approaches for overcoming this challenge.

7.1.4 Class imbalance

UT-Podcast corpus is considered for experimenting with dialect classification. Chapter 3 provides more details on the corpus. From the statistics (AU:449, UK:246, and US:406), it is observed that the classes are imbalanced, and the size of the corpus is very small for investigating it with deep neural networks. With less data and imbalanced classes, the network tends to learn more on the majority class leading to a bias towards a class. Table 7.2 also shows lower performance for the UK due to the bias of the model towards majority classes (AU and US).

Table 7.2: Performance (mean and standard deviation of UAR% from six trials and class-wise accuracies) of CNN classifier for baseline and proposed features.

	Feat. type	**UAR**	**AU**	**UK**	**US**
Baseline features					
STFT-based features	**MFCC-STFT**	57.98±0.54	74.10	11.99	87.85
Proposed features					
ZTW-based features	**ZTWCC**	73.82±1.13	85.94	52.25	83.26
	MFCC-ZTW	67.20±0.20	65.66	47.94	87.99
SFF-based features	**SFFCC**	68.99 ±1.46	81.78	39.14	86.04
	MFCC-SFF	68.15±0.54	72.04	45.32	87.08
FDLP-based features	**FDLPCC**	63.59±5.06	66.95	43.07	80.74

To overcome these class imbalance problem, three approaches were explored for dialect classification in this thesis. They are - (1) class balanced loss (CBL) function, (2) data augmentations, and (3) resampling.

7.1.4.1 Class balanced loss function

To handle the imbalanced classes in the corpus, models are trained with class balanced loss function [142]. This function penalizes the loss for majority classes while providing higher weights for minority classes during training. The loss function with class-balanced weights is expressed as follows:

$$CB(\mathbf{p},y) = \frac{1-\beta}{1-\beta^{n_y}} L(\mathbf{p},y), \tag{7.1}$$

where $\mathbf{p}$ is a vector of class probabilities computed by the classifier given as $[p_1, p_2, \ldots p_C]^T$, y is class label that takes values between 0 to C, and $L(\mathbf{p}, y)$ computed loss. n_y is class strength for class y, $\beta = \frac{N-1}{N}$, and N is the total strength of the corpus.

Results: Table 7.3 shows the performance (mean and standard deviation of UAR% from six trials and mean of class-wise accuracies) for the CNN classifier trained with CBL function for baseline and proposed features. The last column shows the relative improvement with CBL function when compared to Table 7.2.

It can be observed that there is an improvement in performance with CBL function when compared to model trained with an unweighted loss function for both baseline and proposed features. It can also be observed that there is an improvement in the performance of the UK class for all the features.

Table 7.3: Performance (mean and standard deviation of UAR% from six trials and class-wise accuracies) of CNN classifier trained with class balanced loss (CBL) function for baseline and proposed features. RI is relative improvement with class balanced loss function when compared to Table 7.2

	With class balanced loss function					
	Feat. type	**UAR**	**AU**	**UK**	**US**	**RI**
Baseline features						
STFT-based features	**MFCC-STFT**	58.74±1.02	68.69	32.58	70.85	1.31
Proposed features						
ZTW-based features	**ZTWCC**	72.72±0.58	84.65	52.24	84.26	-1.49
	MFCC-ZTW	75.77±0.26	79.18	66.67	81.466	12.75
SFF-based features	**SFFCC**	69.84 ±1.10	78.97	49.87	80.68	1.23
	MFCC-SFF	73.99±0.08	73.566	65.32	83.08	8.57
FDLP-based features	**FDLPCC**	73.03±0.09	78.57	60.49	80.04	14.85

On comparison of MFCC-STFT features with CBL function and without CBL function, it can be observed that there is a relative improvement of 1.31% UAR. For ZTW features, it can be observed that performance of MFCC-ZTW improved by 12.75% UAR (relative) and performance of ZTWCCs remained the same. For SFF features, it can be observed that both SFFCC and MFCC-SFF improved the performance by 1.23% and 8.57 % UAR (relative), respectively. Improvement in performance of UK class can also be observed with SFF features. For FDLP features, it can be observed that FDLPCCs improved the performance by 14.85% UAR (relative) and shown an improvement in UK class accuracy too.

On comparison of the baseline with the proposed features, the proposed features perform better than the baseline features. From the results of ZTW features, ZTWCC and MFCC-ZTW performed better than MFCC-STFT by 23.80% and 28.99% UAR (in relative), respectively. From the results of SFF-based features in the table, it can be observed that SFFCC, and MFCC-SFF outperformed baseline features (MFCC-STFT) by 18.90%, and 25.96% (relative UAR), respectively. FDLPCC features performed better than baseline MFCC-STFT by 24.32% UAR (in relative). Class-balanced loss function have shown greater improvement for proposed features than baseline features.

7.1.4.2 Data augmentation using speed and volume perturbations

Data augmentation is the second approach to overcome the class imbalance problem to generate more data for training using different augmentation approaches such as speed and volume perturbations. Table 7.4 shows the distribution of UT-Podcast corpus before and after data augmentation. The number of utterances available for training in each of the dialects before data augmentation are AU:449, UK:246, and US:406. Data is augmented using speed and volume perturbation approaches to increase the training space, which resulted in, AU:1347, UK:738, and US:1218 utterances. Speed perturbation involves time warping of speech signal $s(t)$ by a factor of α to get $s(\alpha t)$ [125, 143]. Volume perturbation involves the simulation of different recording volumes [125, 144]. Speed perturbation with 0.9 and 1.1 factors and volume perturbation with 1.5 factor resulted in thrice the corpus size. Perturbations are implemented using SoX audio manipulation tool [145].

Table 7.4: Distribution of number of utterances (#utterances) in each dialect class of UT-Podcast (AU: Australian English, UK: Britain English, and US: American English) before data augmentation and after data augmentation for train and test datasets.

	Before data aug.			**After data aug.**		
Data type	AU	UK	US	AU	UK	US
Train	449	246	406	**1347**	**738**	**1218**
Test	332	89	240	332	89	240

Results: DNN architectures are constrained to have sufficiently large amount of data for training. The UT-Podcast dialect corpus used in this study is relatively smaller, and hence different levels of data augmentations (speed, volume, and both) are investigated with CNN classifier. In Table 7.5, Third column (SP) and fourth column (VP) report the results with speed perturbation and volume perturbation respectively, final column (SVP) reports the results with combination of speed and volume perturbations. Experiments

were conducted with baseline feature representations (MFCC-STFT) and proposed feature representations (SFFCC/ZTWCC, MFCC-SFF/MFCC-ZTW, and FDLPCC) to choose the best data augmentation approach for further experiments.

Table 7.5: Performance (mean and standard deviation of UAR% from six trials) of CNN classifier with speed perturbation (SP), with volume perturbation (VP), and with combination of both speed and volume perturbations (SVP).

		After data augmentation		
	Feat. type	**SP**	**VP**	**SVP**
Baseline features				
STFT-based features	**MFCC-STFT**	73.20±0.09	61.91±0.69	76.70±0.56
Proposed features				
ZTW-based features	**ZTWCC**	73.06±0.12	71.81±0.19	74.69±0.14
	MFCC-ZTW	73.92±0.24	75.23±0.46	76.22±1.82
SFF-based features	**SFFCC**	74.42±0.19	73.39±0.34	77.11±0.50
	MFCC-SFF	78.69±0.36	76.61±0.98	76.33±0.68
FDLP-based features	**FDLPCC**	75.16±0.36	76.22±0.40	75.84 ±0.18

The mean and standard deviation of UAR% from six trials are reported in the table. From the standard deviation values, it be can observed that the accuracy is stable across multiple trials. With the individual data augmentation (SP and VP) and combination of data augmentations (SVP), it can be seen that the performance is improved for all the baseline and proposed features.

From the baseline features (row 4), applying speed and volume perturbations individually improved the performance of MFCC-STFT by 26.25% and 6.78% UAR (in relative), and applying both the perturbations together (SVP), improved the performance MFCC-STFT by 32.29% relatively compared to original data.

Applying both the perturbations together (SVP) improved the performance of ZTWCC and MFCC-ZTW by 1.18% and 13.42% (relative UAR), respectively. Independently SP and VP improved the performances of all the SFF-based features. Applying both the perturbations together (SVP) improved the performances of SFFCC and MFCC-SFF by 11.77% and 12.00% (relative UAR), respectively. Independently SP, VP and together SVP improved the performance of FDLPCCs by 18.19%, 19.86%, 19.26% UAR (in relative), respectively. Overall, it can be observed that combination of both speed and volume perturbations (SVP) gave better performance for all the feature representations (baseline and proposed).

Hence the results of combination of speed and volume perturbations data are reported in Table 7.6 in comparison to original data. Table 7.6 shows the performances in UAR (third column) and class-wise accuracies (fourth-sixth columns) with SVP augmentation. The final column (RI) shows the relative improvement with respect to original data (from Table 7.2). It can be observed that along with performance in UAR, the class-wise accuracy of minority class UK improved significantly for all the features. Further, on comparison of performances with CBL function in Table 7.3, performance with SVP is better for both baseline and proposed features.

Table 7.6: Performance (mean and standard deviation of UAR% from six trials and class-wise accuracies) of CNN classifier for baseline and proposed features.

	After speed and volume perturbations					
	Feat. type	**UAR**	**AU**	**UK**	**US**	**RI**
Baseline features						
STFT-based features	**MFCC-STFT**	76.70±0.56	85.07	63.11	81.93	32.29
Proposed features						
ZTW-based features	**ZTWCC**	74.69±0.14	88.52	50.75	85.07	1.18
	MFCC-ZTW	76.22±1.82	64.71	76.97	85.97	13.42
SFF-based features	**SFFCC**	77.11 ± 0.50	85.80	61.42	84.37	11.77
	MFCC-SFF	76.33±0.68	80.46	61.61	87.15	12.00
FDLP-based features	**FDLPCC**	75.84 ±0.18	81.37	61.05	85.3	19.26

7.1.4.3 Resampling

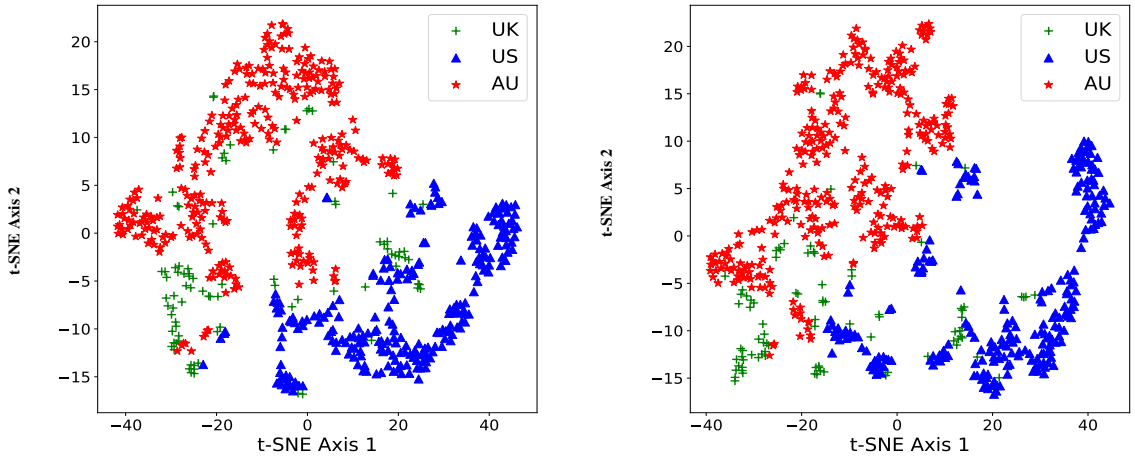
Resampling is the third approach to deal with imbalanced classes [84] in this thesis. It only increases the strength of minority class which helps in reducing the bias of model during training. Table 7.7 shows the distribution of utterances for train and test sets, before and after resampling for UT-Podcast corpus. Resampling only repeat the utterances of minority class UK in training set of UT-Podcast corpus. This change increase the strength of minority class (UK) in training set from 246 to 492.

Discriminability among the dialect classes are visualized using t-SNE projections of latent features. Figure 7.3 shows the t-SNE projections of the latent representations from second fully connected layer of CNN for MFCC-STFT (a) without and (b) with resampling. Projections are color coded by their dialect class (UK:Green(+), US:Blue(Δ), and AU:Red(∗)). It can be observed that all the projections of classes are

Table 7.7: Distribution of number of utterances (#utterances) in each dialect class of UT-Podcast (AU: Australian English, UK: Britain English, and US: American English) before and after resampling for training and test datasets.

	Before resampling			**After resampling**		
Data type	AU	UK	US	AU	UK	US
Train	449	246	406	449	**492**	406
Test	332	89	240	332	89	240

better separated after resampling (Figure 7.3(b)) compared to original data (Figure 7.3(a)). Not only UK class, with resampling AU and US classes are also well separated. These projections are in synchronous with the class-wise accuracies reported in Table 7.8 with CNN.



(a) MFCC-STFT without resampling

(b) MFCC-STFT with resampling

Figure 7.3: Plots showing t-SNE projections of the latent representations from fully connected layer of CNN for MFCC-STFT (a) without and (b) with resampling. Projections are color coded by their dialect class (UK:Green(+), US:Blue(Δ), and AU:Red(∗)).

Table 7.8 shows the performance (mean and standard deviation of UAR% from six trials and mean of class-wise accuracies) for CNN classifier trained with resampled data for baseline and proposed features. Last column shows the relative improvement in perfomances of resampled data function when compared to of original data in Table 7.2.

Overall, there is a significant improvement in performances for both the baseline and proposed features with resampling. The baseline MFCC-STFT features improved by a UAR of 23.46% in relative. From the

Table 7.8: Performance (mean and standard deviation of UAR% from six trials) of CNN classifier for dialect classification with re-sampled corpus. (RI: relative improvement (in %) of re-sample data w.r.t original data (Table 7.2).

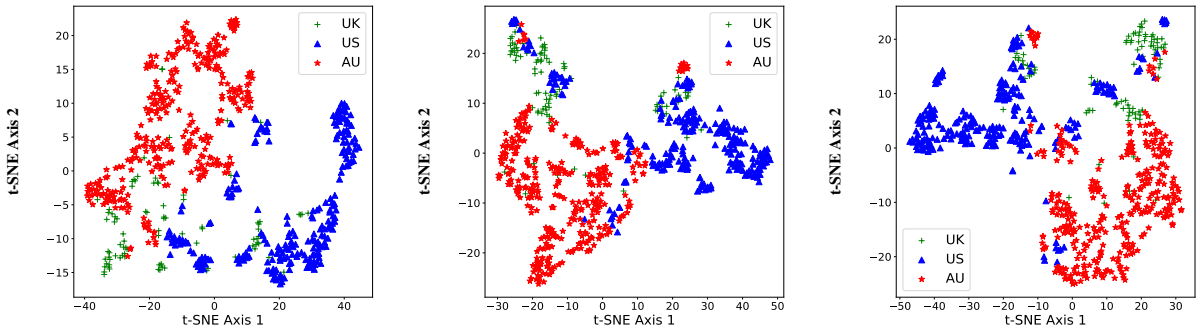
After Re-sampling						
Feat. type		**UAR**	**AU**	**UK**	**US**	**RI**
Baseline features						
STFT-based features	**MFCC-STFT**	**71.58**±0.30	70.18	68.73	76.67	23.46
Proposed features						
ZTW-based features	**ZTWCC**	**78.72**±0.44	79.77	84.27	71.11	6.64
	MFCC-ZTW	**78.33**±0.30	86.30	71.72	76.25	16.56
SFF-based features	**SFFCC**	**79.32**±0.34	87.40	71.35	77.57	14.97
	MFCC-SFF	**80.38**±0.41	87.20	74.91	77.91	17.95
FDLP-based features	**FDLPCC**	**75.24**±0.35	78.87	60.11	86.74	18.32

results of proposed ZTW features, ZTWCC and MFCC-ZTW improved by a UAR of 6.64% and 16.56% UAR (in relative), respectively. From the results of SFF features, SFFCC and MFCC-SFF improved by a UAR of 14.97% and 17.95% UAR (in relative), respectively. FDLPCCs improved by a UAR of 18.32% in relative. When compared to SVP approach, the proposed features with resampling (ZTWCC, MFCC-ZTW, SFFCC, MFCC-SFF, and FDLPCC) have shown significant improvement and MFCC-STFT have shown comparable performance. So, for future experiments with other neural network approaches, resampling is considered.

7.1.5 Comparison of baseline and proposed features with CNN

On comparison of proposed features (ZTW-based, SFF-based, and FDLP-based) with baseline (STFT-based), it is observed that all the proposed features (ZTWCC, MFCC-ZTW, SFFCC, MFCC-SFF, FDLPCC) performed better than baseline as given in Table 7.8. From ZTW-based features, it can be observed that ZTWCC and MFCC-ZTW performed better than MFCC-STFT by a UAR of 9.97% and 9.43% UAR (in relative), respectively. From SFF-based features, it can be observed that SFFCC and MFCC-SFF performed better that MFCC-STFT by a UAR of 10.81% and 12.29% UAR (in relative), respectively. FDLPCCs performed better than MFCC-STFT by 5.11% UAR (in relative). These results are in synchronous with i-vector based system.

Discriminability among the dialect classes are visualized using t-SNE projections of latent features. Figure 7.4 shows the t-SNE projections of the latent features derived from second fully connected layer of CNN classifier for best performing baseline feature (MFCC-STFT: Figure 7.4(a)), proposed SFF-based feature (MFCC-SFF: Figure 7.4(b)), and proposed ZTW-based feature (MFCC-ZTW: Figure 7.4(c)). It can be observed that all the projections of classes are better separated in MFCC-SFF (Figure 7.4(b)) compared to MFCC-STFT (Figure 7.4(a)) and MFCC-ZTW (Figure 7.4(c)). Whereas in Figures 7.4(a) and (c), the projections of classes AU and US are well separated, and the projections of UK class are overlapped with AU and US. Further, these projections are in synchronous with the class-wise accuracies reported in Table 7.8 with CNN.



(a) MFCC-STFT (b) MFCC-ZTW (c) MFCC-SFF

Figure 7.4: Plots showing t-SNE projections of the latent representations from second fully connected layer (FC2, see Section 7.1) of CNN for (a) MFCC-STFT, (b) MFCC-ZTW, and (c) MFCC-SFF. Projections are color coded by their dialect class (AU:Red($*$), UK:Green($+$), and US:Blue(Δ)).

7.2 Time-delay neural network

TDNNs belong to the family of CNNs. TDNN differ from CNNs by introducing sub-sampling in higher layers that led to wider temporal context and doesn't loose much information due to correlated neighbourhood activations. They were first introduced for speech recognition [146] and widely used in extraction of speaker embeddings (x-vectors) [135] and speech recognition [147]. Apart from introducing the wider temporal context, the TDNNs also optimize the time and space complexity during training by reducing the operations (during forward pass and backward propagation) and the parameters of the network.

TDNN differ from CNN because of their sub-sampling. It is a scheme where it will allow selective computations during forward and backward passes reducing the computation complexity. Figures 7.5 and

7.6 demonstrate the wider context of TDNN when compared to CNN. The bottom most row shows the input layer while the last row shows the nodes at third layer. A node in third layer is picked to demonstrate the context captured by that node in both CNN and TDNN cases. Both of them use similar architecture, the kernel or filter size (K) of one dimensional convolution is three with a stride (S) of one for all the layers. In TDNN, sub-sampling or dilation (D) is two across second and third layers. It can be observed that both the networks undergo similar computations, however the node in third layer of TDNN has wider context (five frames from past and five frame from future) than CNN (three frames from past and three frames from future). This shows that the temporal context can be improved with TDNN when compared to CNN.

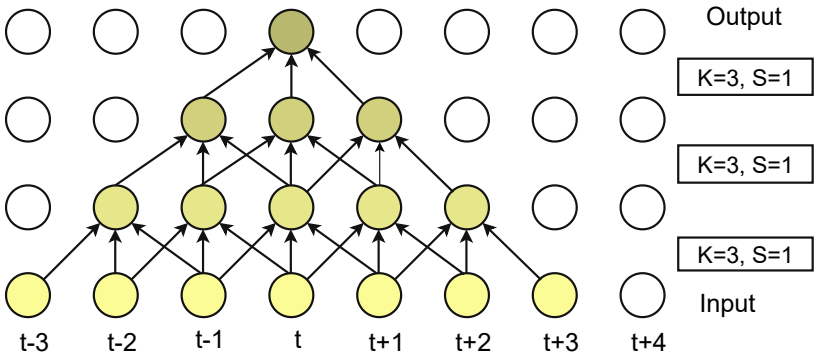


Figure 7.5: Illustration of temporal context in convolution neural network at each layer. Here K is filter size and S is size of sliding window.

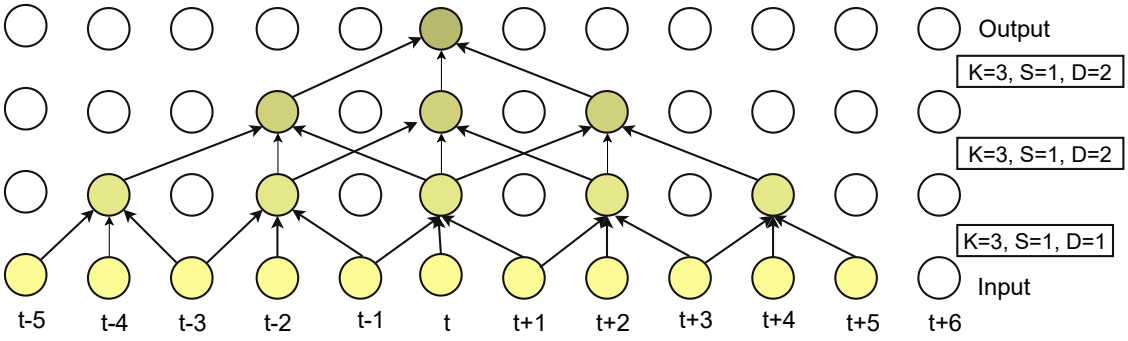


Figure 7.6: Illustration of temporal context in time delay neural network for each layer. Here K is filter size, S is size of sliding window, and D is dilation.

7.2.1 Architecture

Table 7.9 shows the architecture of the TDNN classifier investigated in this study. The time-delay (TD) layers of TDNN are combined with pooling layers and fully connected (FC) layers as in CNNs. The

hyper-parameters that define TD layer are input dimension, output dimension, and context. Along with them cumulative context of the layer is also defined in the table as total context. The first five TD layers process acoustic dependencies at segmental level, while the layers after L2 pooling processes the utterance-level dependencies. The TD layers of TDNN used in this study is similar to the architecture defined in [135] for speaker embeddings.

Table 7.9: End-to-end TDNN architecture for dialect classification. 't' represents current frame and 'T' represents entire utterance. TD represents time-delay layer and FC represents fully connected layer.

Layers:	**TD1**	**TD2**	**TD3**	**TD4**	**TD5**	**L2 pool**	**FC1**	**FC2**	**FC3**
Input dim.	(feat. dim.)*5	1536	1536	512	512	1500T	1500	1500	600
Output dim.	512	512	512	512	1500	1500	1500	600	3
Context	[t-2,t+2]	{t-2,t,t+2}	{t-3,t,t+3}	{t}	{t}	T	0	0	0
Total context	5	9	15	15	15	T	T	T	T

7.2.2 Results and discussion

Table 7.10 shows the performances in mean and standard deviation of UAR from six trials of TDNN classifier for dialect classification with resampled data. The performances are reported for both baseline (MFCC-STFT) and proposed (ZTW, SFF, FDLP based) features with TDNN (in third column) and CNN (in fourth column) classifier. From the table, it can be observed that the performance of all the features with TDNN improved when compared to CNN. It is also observed that all proposed feature performed better than baseline features with TDNN (except FDLPCCs).

From baseline features (MFCC-STFT), it can be observed that with TDNN as classifier the performance of dialect classification improved by 9.97% UAR (relative) when compared to CNN. From ZTW-based features, it can be observed that ZTWCC and MFCC-ZTW features improved by 4.28% and 1.25% UAR (relative), relatively with TDNN when compared to CNN. From SFF-based features with TDNN, it is observed that SFFCCs have shown similar performance and MFCC-SFF have shown an improvement 3.45% UAR (relative) when compared to CNN. FDLPCCs also performed equally well with TDNN when compared to CNN.

On comparison of ZTW based features with MFCC-STFT, it can be observed that, both ZTWCC and MFCC-ZTW performed better with 4.28% and 0.75% UAR (relative), respectively. Among ZTW based features, it is observed that, ZTWCCs performed better than MFCC-ZTW. On comparison of SFF based features with MFCC-STFT, it can be observed that, SFFCC gave a similar performance while MFCC-SFF

Table 7.10: Performance (mean and standard deviation of UAR% from six trials) of TDNN classifier for dialect classification with re-sampled corpus.

	Feat. type	TDNN	CNN
Baseline features			
STFT-based features	**MFCC-STFT**	**78.72±0.84**	71.58±0.30
Proposed features			
ZTW-based features	**ZTWCC**	**82.09±0.62**	78.72±0.44
	MFCC-ZTW	**79.31±0.67**	78.33±0.30
SFF-based features	**SFFCC**	**78.31±0.33**	79.32±0.34
	MFCC-SFF	**83.15±0.76**	80.38±0.41
FDLP-based features	**FDLPCC**	**75.17±0.74**	75.24±0.35

performed better by 5.62% UAR (relative). Among SFF based features, it is observed that, MFCC-SFF features performed better than SFFCCs.

7.3 Temporal convolution neural network

TCNNs [136] belong to the family of CNNs with few constraints. The temporal convolution layers (Tconv) of TCNN differ from CNNs by four architectural changes as given below:

1. Each node of temporal convolution (TConv) layer of the network is constrained only to the past information. This prevents leakage from future to past which is achieved by convolving with k frames in the past (k is the kernel size).

2. TConv layers model sequentially resulting in same output length from each hidden layer. This is achieved by introducing zero-padding of length $(k-1)$ in each hidden layer.

3. The convolutions in each layer are dilated to widen the temporal context without deepening the network. The receptive field at each layer is defined by $(k-1)*d$.

4. Residual block that adds input to output before activation function.

TCNNs were previously explored in speech enhancement for sequential output processing that could replace RNNs with few network parameters and wider context [148]. Motivated by this, TCNs are investigated in classification framework by adding pooling layers and fully connected layers as in CNNs.

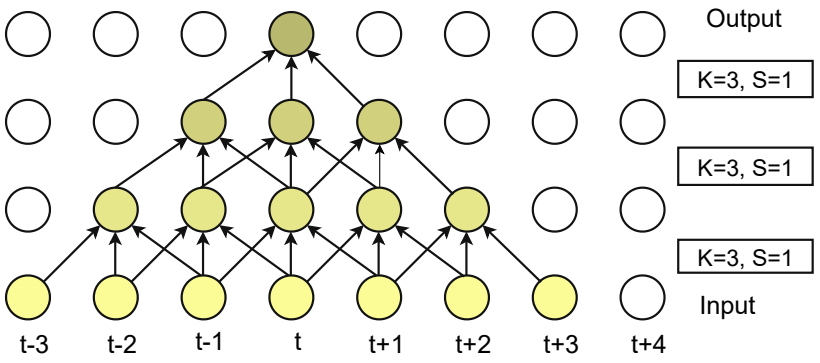


Figure 7.7: Illustration of temporal context in convolution neural network for each layer. Here K is filter size and S is size of sliding window.

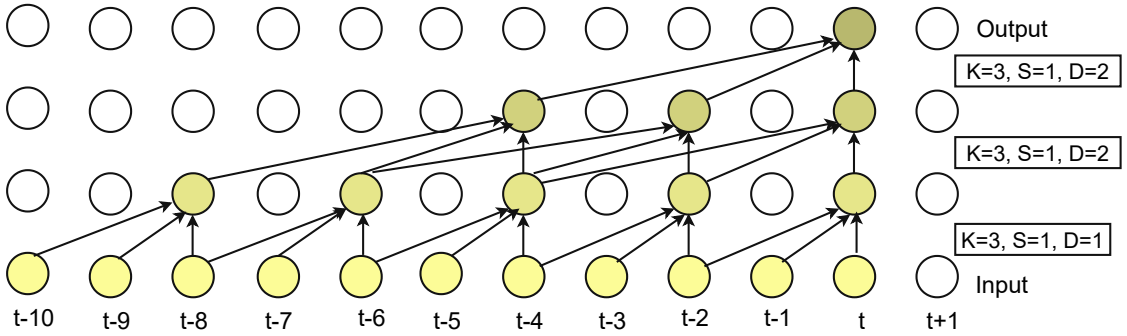


Figure 7.8: Illustration of temporal context in temporal convolution neural network (TCNN) for each layer. Here K is filter size, S is size of sliding window, and D is dilation.

TCNNs differ from CNNs because of their sub-sampling and casual constraint. Sub-sampling, along with casual constraint, improved the temporal context only from past frames. This is demonstrated using Figures 7.7 and 7.8 for CNN and TCNN. The bottommost row shows the input layer, while the last row shows the nodes at the third layer. A node in the third layer is picked to demonstrate the context captured by that node in both CNN and TCNN cases. Both of them use similar architecture, the kernel or filter size (K) of one-dimensional convolution is three with a stride (S) of one for all the layers. In TCNN, sub-sampling or dilation (D) is two across the second and third layers. It can be observed that both networks undergo similar computations. However, the node in the third layer of TCNN has a wider context only from the past (i.e., ten frames from the past) than CNN (three frames from the past and three frames from the future). This shows that the temporal context from the past is improved with TCNN compared to CNN.

7.3.1 Architecture

Table 7.11 shows the architecture of the TCNN classifier investigated in this study. TConv represents the temporal convolution layer. The filters in this network can only access the previous frames with the filter sizes defined in kernel size. The hyperparameters that define the TConv layer are the number of filters (#filters), kernel size, stride, and dilation. After the average pool (L2), the FC layers process the dependencies across the entire utterance.

Table 7.11: End-to-end TCNN architecture for dialect classification. TConv represents the temporal convolution layer, and FC represents the fully connected layer.

Layers:	**TConv1**	**TConv2**	**Max1**	**TConv3**	**TConv4**	**L2 pool**	**FC1**	**FC2**	**FC3**
No. filters/Output dim.	500	80	-	500	500	500	1500	600	3
Kernel size	5	3	10	5	3	-	-	-	-
Stride	1	1	10	1	1	-	-	-	-
Dilation	1	2	-	1	2	-	-	-	-

7.3.2 Results and discussion

Table 7.12 shows the performances in mean and standard deviation of UAR from six trials of TCNN classifier for dialect classification with resampled data. The performances are reported for both baseline (MFCC-STFT) and proposed (ZTW, SFF, FDLP based) features with TCNN (in the third column), TDNN (in the fourth column) and CNN (in the fifth column). From the table, it can be observed that the performance of most of the features improved with TCNN when compared to CNN. It is also observed that all proposed features performed better than baseline features with TCNN (except MFCC-ZTW and FDLPCCs).

On comparison of TCNN with CNN for MFCC-STFT, it is observed that there is an improvement of 9.14% UAR (in relative). ZTW-based features have shown similar performance for TCNN and CNN. From SFF-based features, it can be observed that SFFCCs improved the performance by 0.32% UAR (relative) and MFCC-SFFs gave a similar performance for classification. With TCNNs, FDLPCCs performed better by 0.66% UAR (relative). Overall, comparing three different neural networks (CNN, TDNN, and TCNN), it can be observed that TDNN performed better for most of the features.

On comparison of proposed features (ZTWCC, MFCC-ZTW, SFFCC, MFCC-SFF, and FDLPCC) to baseline MFCC-STFT features with TCNN, it is observed that ZTWCC, SFFCC, and MFCC-SFF performed better by 0.70%, 1.93%, 3.29% UAR (relative), respectively. Overall, the proposed MFCC-SFF with TCNN performed with a UAR of 80.69%.

Table 7.12: Performance (mean and standard deviation of UAR% from six trials) of TCNN classifier for dialect classification with re-sampled corpus. Performances of CNN and TDNN classifiers are also reported.

	Feat. type	TCNN	TDNN	CNN
Baseline features				
STFT-based features	**MFCC-STFT**	**78.12±0.57**	78.72±0.84	71.58±0.30
Proposed features				
ZTW-based features	**ZTWCC**	78.67± 1.67	82.09±0.62	78.72±0.44
	MFCC-ZTW	76.17±0.93	79.31±0.67	78.33±0.30
SFF-based features	**SFFCC**	**79.63 ± 0.88**	78.31±0.33	79.32±0.34
	MFCC-SFF	**80.69 ± 0.96**	83.15±0.76	80.38±0.41
FDLP-based features	**FDLPCC**	**75.74±1.36**	75.17±0.74	75.24±0.35

7.4 Emphasized channel attention, propagation and aggregation in TDNN

From the experimental observations in previous sections, it is observed that TDNN has shown better performance for dialect classification. So, multiple enhancements were made to TDNN which resulted in ECAPA-TDNN [137]. They are mainly by introducing the following modules with TDNN: (1) **Squeeze-Excitation Res2Block** scales each channel based on global knowledge. (2) **Multilayer feature aggregation and summation**, which captures the relevant information from both shallow and deeper feature maps. (3) **Channel and context-dependent statistics pooling** computes both temporal and channel attention weighted mean and standard deviation.

- **Squeeze-Excitation Res2Block** (SE Res2Block) combines the benefits of Squeeze-Excitation (SE) block (scales each channel according to global properties of the utterance) with Res2Net module (computes multi-scale features with hierarchical residual connections within and reduces the model parameters) [149]. Equation 7.4 shows the computation of SE block. Channel weight s_c scales down each channel based on global mean descriptor z.

$$\mathbf{z} = \frac{1}{T}\sum_{t}^{T}\mathbf{h_t} \tag{7.2}$$

$$\mathbf{s} = \sigma(\mathbf{W_2} f(\mathbf{W_1 z} + \mathbf{b}_1) + \mathbf{b}_2) \tag{7.3}$$

$$\bar{\mathbf{h_c}} = s_c \mathbf{h_c} \tag{7.4}$$

The SE Res2Block contains a time-delay layer which is preceded and succeeded by convolution layer which is then followed by a SE block. This entire block has a residual connection over it.

- **Multilayer feature aggregation and summation** of feature maps from all three SE Res2Blocks. The shallow feature maps can also contribute towards more robust dialect embeddings. To capture the relevant information from both shallow and deeper feature maps, the outputs from three SE-Res2Blocks are aggregated and summated.

- **Channel and context-dependent statistics pooling** is used to convert variable-length frame-level features to fixed-length utterance-level features. This module is also called attentive statistic pooling (ASP). Mean and standard deviations are computed by the weights computed based on both temporal and channel attention scores. The temporal and channel attention scores are computed as follows:

$$e_{t,c} = \mathbf{v_c^T} f(\mathbf{W h_t} + \mathbf{b}) + k_c \tag{7.5}$$

$$\alpha_{t,c} = \frac{\exp e_{t,c}}{\sum_{\tau}^{T} \exp e_{\tau}} \tag{7.6}$$

With $\alpha_{t,c}$ as temporal and channel scores, weighted mean vector is given as follows:

$$\mu_c = \sum_{t}^{T} \alpha_{t,c} \mathbf{h_{t,c}} \tag{7.7}$$

Higher-order statistics (i.e., standard deviations as utterance-level features) are effective for higher discriminability. So, the mean is concatenated with the standard deviation as fixed utterance level representations. Standard deviation is computed as:

$$\sigma_c = \sqrt{\sum_{t}^{T} \alpha_{t,c} \mathbf{h_{t,c}}^2 - \mu_c^2} \tag{7.8}$$

7.4.1 Architecture

Figure 7.9 shows the architecture of ECAPA-TDNN studied in this thesis, which is similar to that in [138]. One dimensional convolution layer (Conv1D) is the input layer. Followed by three layers of SE-Res2Blocks are sequentially arranged. Each SE-Res2Block is elaborated on the right side of the figure. Each SE-Res2Block has 2 Conv1D layers with a time-delay layer in between. This is followed by an SE block which scales down the channel based on global characteristics. A skip connection is over the entire block to reduce gradient degradation problem. The Output from three SE-Res2Blocks are concatenated using aggregation and summation. After this aggregation, a Conv1D layer processes the concatenated information to generate the features for the attentive statistics pooling (ASP). The weighted mean and standard deviation are computed in attentive statistics pooling (ASP) layer. These statistics are computed

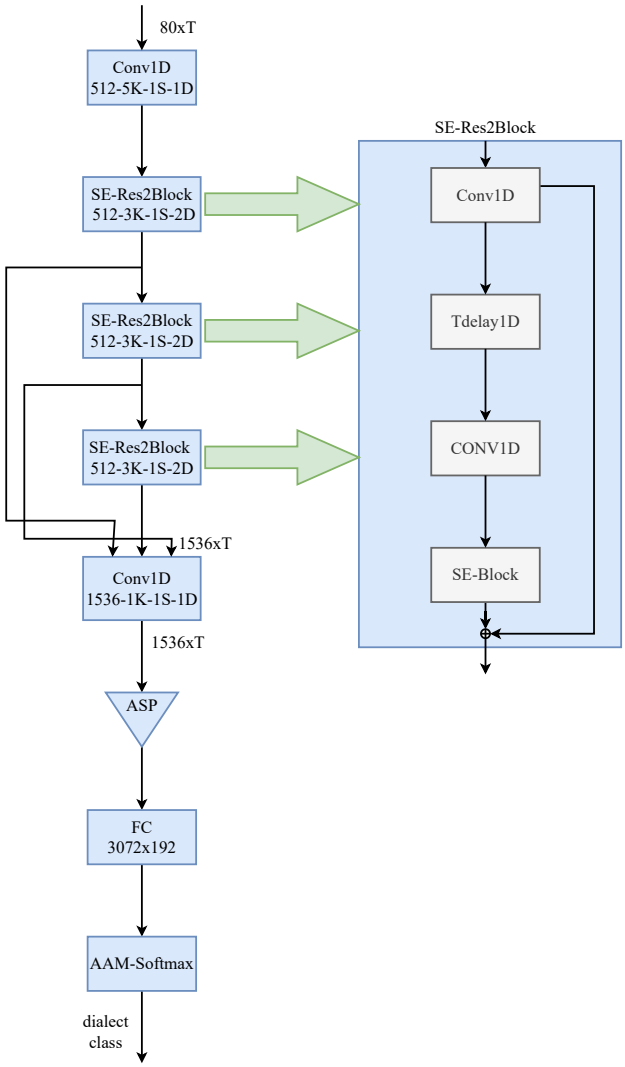


Figure 7.9: A schematic block diagram showing architecture of ECAPA-TDNN. ASP: Attentive statistic pooling, Conv1D:1 dimensional convolution layer, FC: fully connected feed forward layer.

using both channel and context attention weights and these statistics represent utterance-level features. Finally, fully connected (FC) layer with softmax layer is used to classify dialect.

7.4.2 Results and discussion

Table 7.13 shows the performances in mean and standard deviation of UAR from six trials of ECAPA-TDNN classifier for dialect classification with resampled data. The performances are reported for both baseline (MFCC-STFT) and proposed (ZTW, SFF, FDLP based) features with ECAPA-TDNN (in the third column), TCNN (in the fourth column), TDNN (in the fifth column), and CNN (in last column). From

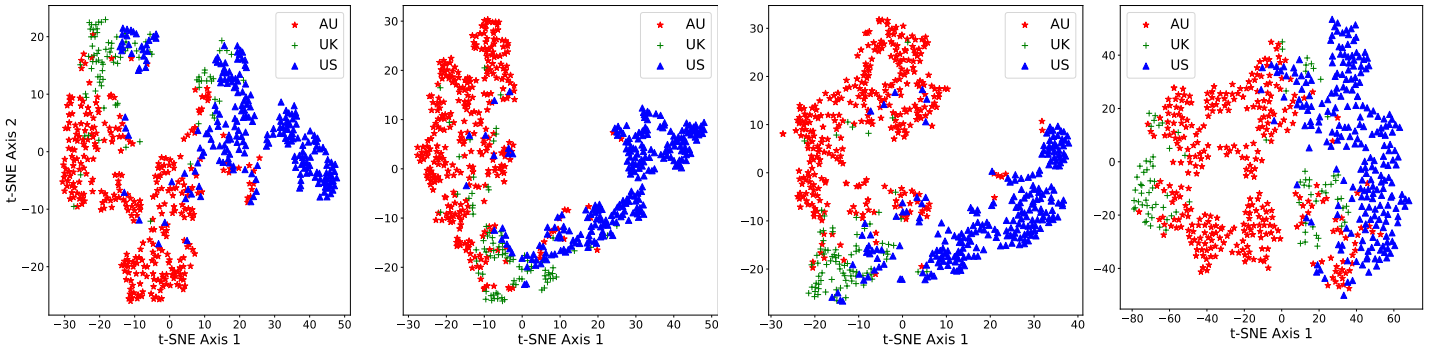
the table, it can be observed that the performance of all the features improved with ECAPA-TDNN when compared to CNN. It is also observed that all proposed features performed better than baseline features.

From baseline features (MFCC-STFT), it can be observed that with ECAPA-TDNN, the performance of dialect classification improved by 13.52% UAR (relative) compared to CNN. Among ZTW-based features, ZTWCC and MFCC-ZTW features improved by 4.46% and 4.43% UAR (relative), relatively with ECAPA-TDNN compared to CNN. Among SFF-based features, SFFCC and MFCC-SFF features improved by 6.33% and 3.53% UAR (relative), relatively with ECAPA-TDNN compared to CNN. FDLPCCs also improved their performance with ECAPA-TDNN compared to CNN by 8.79% UAR (relative).

Table 7.13: Performance (mean and standard deviation of UAR% from six trials) of ECAPA-TDNN classifier for dialect classification with re-sampled corpus. Performances of CNN, TDNN, and TCNN classifiers are also reported.

	Feat. type	**ECAPA**	**TDNN**	**TCNN**	**CNN**
Baseline features					
STFT-based feat.	**MFCC-STFT**	**81.26±1.46**	78.72±0.84	78.12±0.57	71.58±0.30
Proposed features					
ZTW-based features	**ZTWCC**	**82.23±2.16**	82.09±0.62	78.67± 1.67	78.72±0.44
	MFCC-ZTW	**81.80±1.12**	79.31±0.67	76.17±0.93	78.33±0.30
SFF-based features	**SFFCC**	**84.34±2.12**	78.31±0.33	79.63 ± 0.88	79.32±0.34
	MFCC-SFF	**83.22±0.98**	83.15±0.76	80.69 ± 0.96	80.38±0.41
FDLP-based features	**FDLPCC**	**81.85±0.95**	75.17±0.74	75.74±1.36	75.24±0.35

Comparing ZTW-based features to MFCC-STFT with ECAPA-TDNN, it can be observed that both ZTWCC and MFCC-ZTW performed better by 1.19% and 0.66% UAR (relative), respectively. Among ZTW-based features, it is observed that ZTWCCs performed better than MFCC-ZTW. Comparing SFF-based features to MFCC-STFT with ECAPA-TDNN, it can be observed that both SFFCC and MFCC-SFF performed better by 3.79% and 2.41% UAR (relative), respectively. Among SFF-based features, it is observed that SFFCCs performed better than MFCC-SFF. The performance of FDLPCCs is better when compared to MFCC-STFT with ECAPA-TDNN by 0.73% UAR (relative).



(a) CNN (b) TCNN (c) TDNN (d) ECAPA-TDNN

Figure 7.10: Plots showing t-SNE projections of the latent representations from second fully connected layer of CNN (a), TCNN(b), TDNN (c), and ECAPA-TDNN (d) for SFFCC features. Projections are color coded by their dialect class (AU:Red($*$), UK:Green($+$), and US:Blue(Δ)).

Figure 7.10 shows the t-SNE projections of the latent features derived from four deep neural classifiers, CNN (Figure 7.10(a)), TCNN(Figure 7.10(b)), TDNN (Figure 7.10(c)), and ECAPA-TDNN (Figure 7.10(d)) trained with one of proposed features (SFFCCs). From t-SNE projections of CNN (Figure 7.10(a)), it can be observed that the projections of classes AU and US are well separated, and the projections of UK class are overlapped with AU and US. Whereas from t-SNE projections of TCNN(Figure 7.10(b)), TDNN (Figure 7.10(c)), and ECAPA-TDNN (Figure 7.10(d)), all the classes are relatively better separated when compared to CNN. These observations are in conformity with the class-wise accuracies reported in Table 7.13 for SFFCC features.

7.5 Results and discussion

After analysis of experiments without and with data augmentations to handle imbalanced classes with CNN, investigation of different temporal contexts with CNN architecture, and investigation of different DNN architectures as given in previous sections. This subsection briefly reports the experiments that compared only the best performing DNN (ECAPA-TDNN) to i-vector based system for best performing baseline and proposed features (MFCC-STFT, MFCC-ZTW, MFCC-SFF, and FDLPCC). Further, this section also reports the comparison (best performing DNN (ECAPA-TDNN) with baseline and proposed features (MFCC-STFT, MFCC-ZTW, MFCC-SFF, and FDLPCC)) to previous studies with DNN for dialect classification.

7.5.1 Comparison to i-vector based dialect classification system

Table 7.14 shows the performance (in UAR%) of i-vector based dialect classification system (Chapters 4-6) and performance (in mean and standard deviation of UAR% from six trials) for best performing ECAPA-TDNN with baseline (STFT-based) and proposed (SFF, ZTW, and FDLP-based) features. From table, it is observed that both the baseline (MFCC-STFT) and proposed (MFCC-ZTW, MFCC-SFF, and MFCC-ZTW) features improved when compared to i-vector system. Among all the features, MFCC-SFF based features with ECAPA-TDNN performed better. It is also observed that the minor class UK performed well (with accuracy $> 70\%$) for all proposed features (MFCC-ZTW, MFCC-SFF, and FDLPCC) with deep neural architecture (ECAPA-TDNN).

Table 7.14: Performance (in UAR%) of i-vector system (with original UT-Podcast) and performance (in mean and standard deviation of UAR% from six trials) for best neural network architecture (ECAPA-TDNN) with baseline (STFT-based) and proposed (SFF, ZTW, and FDLP-based) features (with resampled UT-Podcast).

Feat. type	**UAR**	AU	UK	US
i-vector system (Chapters 4 - 6)				
MFCC-STFT (baseline)	**77.98**	87.35	56.18	90.42
MFCC-ZTW (proposed)	**78.73**	87.35	58.43	90.42
MFCC-SFF (proposed)	**81.20**	**97.59**	53.93	**92.08**
FDLPCC (proposed)	**81.37**	**86.14**	**66.29**	**91.67**
ECAPA-TDNN Architecture				
MFCC-STFT	**81.26±1.46**	82.38	69.10	92.29
MFCC-ZTW	**81.80±1.12**	75.10	78.84	91.46
MFCC-SFF	**83.22±0.98**	79.12	79.78	90.76
FDLPCC	**81.85±0.95**	86.3	71.35	87.92

Figure 7.11 shows confusion matrices for dialect classification with i-vectors and ECAPA-TDNN for baseline (MFCC-STFT) and proposed (MFCC-ZTW, MFCC-SFF, and FDLPCC) features. Each value represents the rate of a samples belonging to an actual class predicted as resultant class in %. The rows represent actual class values, so the rows add up to 100. The value along diagonal shows the class-wise accuracies while the other values gives the confusion percentage of actual class with predicted class. On

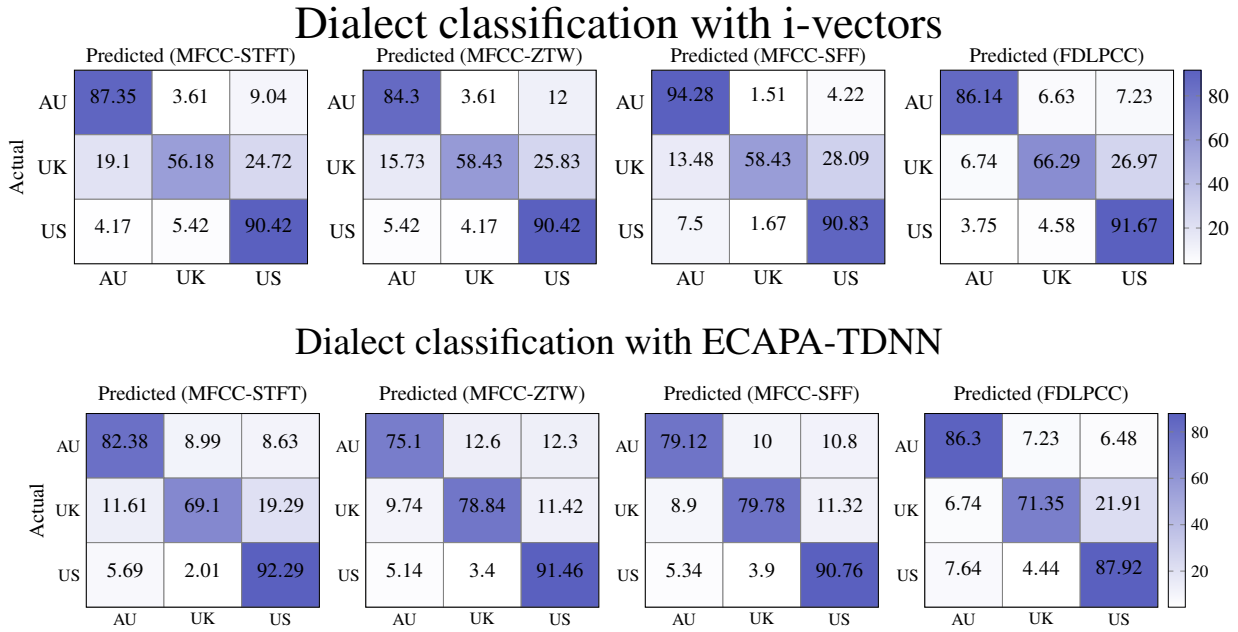


Figure 7.11: Confusion Matrices for dialect classification system with i-vector and ECAPA-TDNN systems for both baseline (MFCC-STFT) and proposed (MFCC-ZTW, MFCC-SFF, and FDLPCC) features.

comparison of i-vectors to ECAPA-TDNN, it can be observed that both baseline and proposed features improved in UK class accuracy. It can also be observed that both confusion of UK class to AU and US (values in first and third columns of second row) have reduced with ECAPA-TDNN when compared to i-vectors for all features. Further, it is observed that MFCC-SFF have shown lower confusion across all the matrix.

7.5.2 Comparison with previous studies

This section compares the results obtained for UT-Podcast corpus by the previous approaches [84] that uses DNNs and the current studies (with both baseline and proposed features). In the previous study [84], the strength of utterances belonging to minority class (UK) are re-sampled for training. They investigated six different neural architectures (feed-forward neural network (FFNN), five-layer CNN, AlexNet, VGG-11, ResNet-18 and FreqCNN) with STFT spectrogram as input. Feed-forward neural network is a small deep neural classifier with three fully connected layers. Five-layer CNN is a deep neural classifier with five 2D convolution layers followed by fully connected layers. AlexNet [129], VGG-11 [130], and ResNet [131] are typical deep neural architectures belong to family of CNNs with varied number of convolution layers. FreqCNN is proposed in [84], and it's architecture comprises of attention based convolution blocks along with basic convolution blocks.

Table 7.15 shows the results (UAR and class-wise accuracies) from previous studies in [84] that uses different neural networks with SPEC-STFT as input, and the results of proposed and baseline features with ECAPA-TDNN classifier. The UAR% and class-wise accuracies of the current studies are the mean values from six trials. For brief discussions, only the MFCC-STFT from baseline and MFCC-ZTW, MFCC-SFF, and FDLPCC from proposed are considered for comparison from current study. Among the six different DNNs from previous studies [84], it can be observed that FreqCNN performed better (with 79.32% UAR) than other classifiers. On the other hand, it can be observed that current studies with all the baseline and proposed features (especially MFCC-SFF features) performed better than the previous studies. From the comparison of class-wise accuracies among previous studies, it can be observed that other than AlexNet and FreqCNN, all the classifiers identified UK dialect with less than 50%. However, AlexNet lacked its performance in identifying AU dialect. On the other hand, almost all the proposed features identified UK dialects with accuracy more than 70% without lacking performance in other dialect classes (AU and US). Current studies with both baseline and proposed features outperformed all the architectures of previous studies with similar data configurations.

Table 7.15: Performance in UAR% (mean and standard deviation from six trials) and class-wise accuracies (of classes AU, UK, and US) for different deep neural architectures from previous studies and current studies with all the features (STFT, ZTW, SFF, and FDLP based) using best DNN architecture (ECAPA-TDNN) (with resampled UT-Podcast).

			Class-wise accuracies		
Input Feat. Type	**Arch. type**	**UAR**	**AU**	**UK**	**US**
Previous studies [84]					
SPEC-STFT	FFNN	61.42	70.78	50.56	62.92
	Five-layer CNN	62.81	64.76	41.57	82.0
	AlexNet	**64.90**	58.43	64.04	74.17
	VGG-11	54.40	55.72	48.31	59.17
	ResNet-18	61.66	69.28	38.20	77.50
	FreqCNN	**79.32**	88.55	71.91	77.50
Current studies					
MFCC-STFT	ECAPA-TDNN	**81.26±1.46**	82.38	69.10	92.29
MFCC-ZTW		**81.80±1.12**	75.10	78.84	91.46
MFCC-SFF		**83.22±0.98**	79.12	79.78	90.76
FDLPCC		**81.85±0.95**	86.3	71.35	87.92

Table 7.15 shows the results (UAR and class-wise accuracies) from previous studies in [84] and the results of baseline and proposed features with three end-to-end classifiers from current studies. For comparison with previous studies, results obtained for baseline (MFCC-STFT) and proposed (MFCC-ZTW, MFCC-SFF, and FDLPCC) features are considered as in Table 7.13. Among the five different DNNs from previous studies [84], it can be observed that FreqCNN performed better than other classifiers. On the other hand, it can be observed that all the proposed features with all the end-to-end classifiers outperformed the previous studies.

7.6 Summary and conclusions

Major goal of this thesis is to study the dialectal variations and improve the performance of speech recognition with an improved dialect classification system. So, initial studies proposed to use advanced signal processing approaches that discriminate dialects better with traditional i-vector system for dialect classification system. Then, based on our observations, basic to advanced deep neural networks are investigated with proposed features for dialect classification to get best out of both stages. These approaches are investigated with major dialects of English (AU, UK, and US). In most studies, Indian English is considered as single dialect even though it has different native speakers. Based on the conclusions made from major dialects of English (AU, UK, and US), embeddings from improved dialect classification system are included with Indian English ASR to improve the performance.

Proposed features that provide high spectral resolution without compromising temporal resolution derived using advanced signal processing approaches such as ZTW and SFF are investigated. It is observed that proposed features (SFFCC, MFCC-SFF, ZTWCC, and MFCC-ZTW) performed better than baseline features. Further, experimentation with features that provide longer temporal summarization derived using FDLP based features are investigated for dialect classification. Proposed features are investigated for different cepstral orders and temporal contexts for dialect classification. From the experiments, it is observed that proposed features (FDLPCC) performed better than baseline features.

From the experiments with simpler (CNN) to advanced deep neural network (TCN, TDNN, and ECAPA-TDNN) architectures that provide different temporal contexts, it is observed that advanced neural network architectures improved the performance of dialect classification. It is also observed that proposed features derived from SFF performed better. From the experiments, it is observed that DNN based ECAPA-TDNN performed better in dialect classification than i-vector based approach. With ECAPA-TDNN, the proposed MFCC-ZTW, MFCC-SFF, and FDLPCC outperformed the MFCC-STFT by 0.66%, 2.41%, and 0.73% (relative UAR), respectively. The best performance is given by MFCC-SFF with ECAPA-TDNN architecture.

Further, the best embeddings derived from improved dialect classification system are applied in English ASR system in the following chapters. Furthermore, in extension to Indian English (with different L1 speakers), a dialect classification system with ECAPA-TDNN architecture trained with SFF based features will be developed to derive dialect embeddings. These embeddings will be included in Indian English ASR to handle dialectal variations.

Chapter 8

Leveraging dialect embeddings in multi-dialect ASR system

Dialectal variations in speech can influence the performance of speech recognition systems [8–10]. In [8], cross-accent models were found to increase the error rate by 40-50% (relative) compared to accent-specific models. To deal with such situations, three different solutions were presented in the literature. They are, multi-accented acoustic model [8, 9], accent dependent acoustic model [8], and accent adaptation methods [9, 12].

In the multi-accented acoustic model approach, automatic speech recognition (ASR) models were trained with multiple accents of data to learn the traits of all the accents collectively [13]. Developing accent-dependent models requires more significant amounts of data (from each dialect) for training each model independently [8, 14].

In acoustic model adaptations, various techniques such as MAP/MLLR adaptation of traditional HMM-GMM acoustic model [15, 16], fine-tuning acoustic models to specific accents [13, 14], and inclusion of accent embeddings [9, 10] were carried out to improve the performance. Joint modelling of accent recognizer and speech recognizer was shown to improve the performance of the ASR system with seen accented and unseen accented data [9, 12].

8.1 Multi-dialect speech recognition architectures

This section discusses five different architectures of DeepSpeech2 [150] system that evaluate multi-dialect ASR system.

8.1.0.1 Pre-trained DeepSpeech2 model

DeepSpeech2 model is an end-to-end ASR model that maps a sequence of input features to a sequence of graphemes [150]. Connectionist Temporal Classification (CTC) loss [151] is used to train the network. Figure 8.1 (without L1 embeddings block) shows the block diagram of DeepSpeech2 architecture. Short-time Fourier transform (STFT) spectrogram from a speech signal is computed with a 20 msec

Hamming window with a shift of 10 ms. A sequence of 161-dimensional STFT spectrogram is passed through two convolution 2D (Conv2D) layers with 32 filters of sizes (41,11) and (21,11). A stride of 2 along frequency in the first and second layers results in 41-dimensional features from 32 filters. Flattening them results in 1312-dimensional features per time frame, which is then passed through 5 bi-directional long short-term memory (LSTM) models. The output of LSTM at each time frame is passed to a fully connected (FC) layer with a softmax function to predict the grapheme corresponding to that time frame.

Pre-trained DeepSpeech2 model is the model obtained after training the DeepSpeech2 model with 960 hrs of US-accented Librispeech corpus [152]. The pre-trained system is evaluated for each dialect.

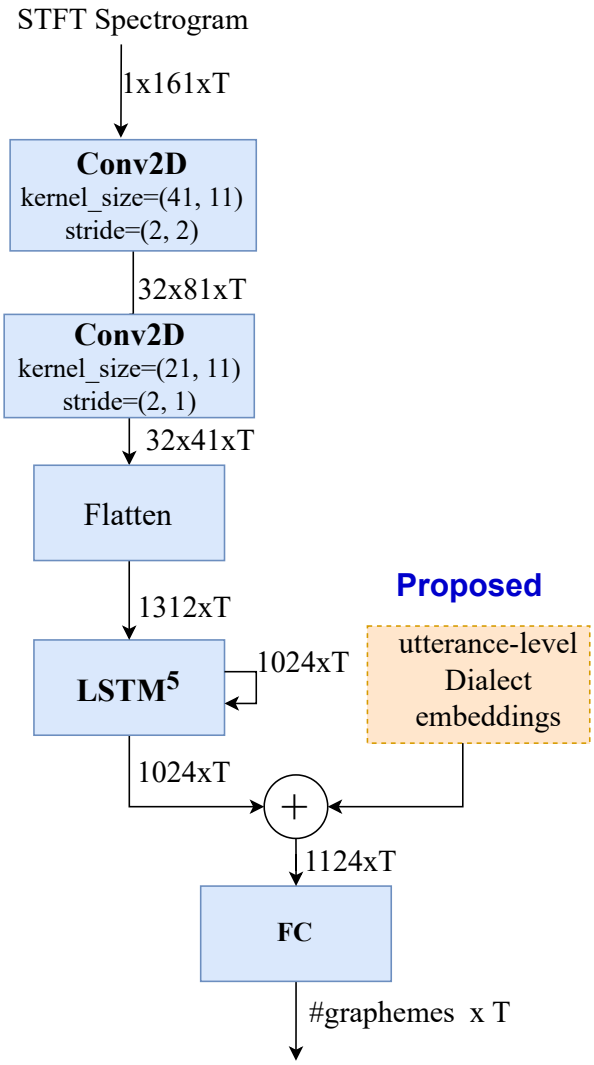


Figure 8.1: Block diagram of end-to-end DeepSpeech2 architecture with proposed utterance-level dialect embeddings.

8.1.0.2 Fine-tuned DeepSpeech2 model

The pre-trained DeepSpeech2 model is fine-tuned with the training data from the respective corpus to improve speech recognition performance. This way the model can learn all the external (environment) and internal (speaker) variabilities in speech. With the corpus containing data from all the dialects, the model will be fine-tuned to all dialectal traits as well.

8.1.0.3 L1 embeddings with DeepSpeech2 model

The inclusion of dialect embeddings that contain the pronunciation traits to differentiate dialects can improve the performance of multi-dialect ASR system. The i-vector based dialect embeddings and ECAPA-TDNN based dialect embeddings extracted from both STFT based features and SFF based features are investigated for speech recognition in this chapter.

Figure 8.1 shows the proposed DeepSpeech2 architecture with utterance-level dialect embeddings. The 100-dimensional i-vectors/ECAPA-TDNN based dialect embeddings are concatenated with the output (of size 1024) of fifth LSTM layer to obtain 1124xT matrix (T denotes total number of time frames). Only the FC layer of the DeepSpeech2 network is trained to learn the dialectal traits.

The inclusion of L1 embeddings that contain the accentual traits to differentiate L1 accent classes can improve the performance of ASR with Indian English (L2).

Multi-task joint learning with similar to the architecture as in [9] but with RNNs as in deep speech2 is also experimented. ASR is trained jointly with an objective $L_{joint} = \alpha * L_{AM} + (1-\alpha) * L_{DID}$ with α as 0.8.

8.1.0.4 Combining i-vectors and ECAPA-TDNN embeddings with DeepSpeech2

The i-vector dialect embeddings obtained in an unsupervised manner, not only contain dialect information but also other variant information (speaker, gender, etc.) and ECAPA-TDNN embeddings are trained to contain only dialect embeddings. So, both the embeddings (i-vector based dialect embeddings and ECAPA-TDNN based dialect embeddings) are combined, which forms 200-dimensional utterance-level dialect embeddings. The 200-dimensional dialect embeddings are concatenated with the output (of size 1024) of fifth LSTM layer to obtain 1224xT matrix. Similar to the above, only the FC layer of the DeepSpeech2 network is trained to learn the dialect traits with dialect embeddings derived from both MFCC-STFT and MFCC-SFF features.

8.2 Leveraging dialect embeddings in speech recognition system

The multi-dialect ASR system trained on American English (US) is investigated for three major dialects (American: US, Australian: AU, and Britian: UK). The embeddings derived from an improved dialect

classification system with major dialects of English (AU, UK, and US) are leveraged with a multi-dialect ASR system. Common voice corpus [153] with three major dialects is considered to evaluate multi-dialect ASR system. Based on the observations with UT-Podcast, the embeddings derived from the best dialect classification model (ECAPA-TDNN with MFCC-SFF) will be applied to the speech recognizer.

8.2.1 Results and discussion

Table 8.1 shows the performance (in WER%) of ASR systems (pre-trained, fine-tuned, i-vector based dialect embeddings, ECAPA-TDNN based dialect embeddings, and combined dialect embeddings) for the major dialects (AU, UK, and US) of English. In addition, to dialect-wise WER (columns 2-4) and average WER (column 5), the relative improvement (Rel. imp.) with respect to the pre-trained model and fine-tuned models are reported in the table. The pre-trained model (Section 8.1.0.1) is evaluated for speech from AU, UK, and US dialects. From the results, it can be observed that the US dialect performed better compared to AU and UK, with a WER% of 22.67. As the pre-trained model is trained on the speech from the US dialect, the performance of the US is better than the other dialects is admissible. Between AU and UK dialects, the UK performed better than AU.

The pre-trained DeepSpeech2 model is fine-tuned with all three dialects (Section 8.1.0.2) with around 24.5 hrs of common voice corpus. The test data is excluded from the training data used for fine-tuning. The fifth column of the table shows the performance (in WER%) of fine-tuned speech recognition system. It can be observed that, on an average, fine-tuning improved the performance of pre-trained by 10.07% WER (relative). The recognition performance of all three dialects improved with the fine-tuned model.

Dialect embeddings derived from i-vector and ECAPA-TDNN dialect classification systems. These embeddings are used in finetuned models as in Section 8.1.0.3. With MFCC-STFT features in i-vector and ECAPA-TDNN systems, both the embeddings improved the performance of the ASR system. Relatively i-vector based embeddings improved by 10.46% and 0.44% WER when compared to pre-trained and fine-tuned DeepSpeech2 models, respectively. Additionally with joint training of ASR and dialect classification, it can be observed that joint training of ASR and DID have shown to be slightly better than dialect embeddings.

With SFF based i-vector dialect embeddings, the performance of the ASR system improved by 11.08% WER and 1.13% WER (relative). While using SFF based ECAPA-TDNN dialect embeddings, improved by 11.52% and 1.61% WER (relative). It can also be observed that by leveraging SFF dialect embeddings there is a slight improvement in WER when compared to STFT dialect embeddings.

The i-vector embeddings and ECAPA-TDNN embeddings are combined and are leveraged in the DeepSpeech2 model as in Section 8.1.0.4. The last two rows of the table show the performance in

Table 8.1: Performance (in WER%) of ASR systems (pre-trained, fine-tuned, i-vector based dialect embeddings, ECAPA-TDNN based dialect embeddings, and combined dialect embeddings) for major dialects of English. Rel. imp. refers to relative improvement.

Model Type	**AU**	**UK**	**US**	**Average**	**Rel. imp. w.r.t. pre-trained**	**Rel. imp. w.r.t. fine-tuned**
Pre-trained DeepSpeech2						
Pre-trained DeepSpeech2	28.57	25.36	22.67	25.53	-	-
Fine-tuned DeepSpeech2						
Fine-tuned DeepSpeech2	26.52	22.77	19.58	22.96	10.07	-
STFT dialect embeddings with Fine-tuned DeepSpeech2						
i-vector based dialect emb.	26.52	22.34	19.72	22.86	10.46	0.44
ECAPA-TDNN based dialect emb.	26.28	22.19	19.91	22.79	10.73	0.74
Joint training [ASR+DID]	26.48	22.36	19.42	22.75	10.89	0.91
SFF dialect embeddings with Fine-tuned DeepSpeech2						
i-vector based dialect emb.	26.60	22.26	19.25	22.70	11.08	1.13
ECAPA-TDNN based dialect emb.	26.50	22.16	19.12	22.59	11.52	1.61
Combined dialect embeddings with DeepSpeech2						
STFT i-vector+ECAPA-TDNN emb.	27.16	21.53	19.10	22.60	11.48	1.57
SFF i-vector+ECAPA-TDNN emb.	26.32	22.10	19.02	22.48	11.95	2.09

WER% with combined dialect embeddings. From the experiments, it can be concluded that using SFF i-vector+ECAPA-TDNN embeddings performed better than any other with a performance of 22.48% WER.

It can be understood that dialect embeddings are helpful for speech recognizers with multi-dialect speech. Further, the improvement of MFCC-SFF embeddings in multi-dialect ASR system is relative to its better performance in classification.

8.3 L1 identification and leveraging L1 embeddings in Indian English ASR system

The native language (L1) traits can be observed in the non-native (L2) speech of a speaker. The influence of the native language of the speaker may lead to mispronunciations. These mispronunciations of non-native speakers may lead to the misrecognition of words leading to higher word error rates.

These mispronunciations are due to the effect of native language (L1) phonology on second language speech (i.e., non-native speech (L2)). In most of the existing studies, Indian English is considered as one class, even though the speech is multi-lingual [154]. From previous studies with acoustic model adaptation [9,10], it was observed that the WER of Indian English was higher than other accents by a margin of 10-20% (absolute). Jointly training ASR systems for L2 speech (Indian English) and L1 speech (Hindi, Kannada, Gujarati, Marathi, Tamil, and Telugu) has shown to improve the performance [155]. These improvements suggest that L1 influences L2 speech (Indian English). From the linguistic studies in [154, 156, 157], it was found that there exists some effect of L1 on Indian English, even though the degree of this effect varies due to multiple factors. Authors in [14] divided Indian English into sub-groups (North, East, West, and South) based on the ASR performance on cross L1 accent models. Note that in [14], Kannada, Malayalam, Tamil, and Telugu are considered as one sub-group, as all of them are south Indian languages. The present study considers Indian English from five closely related, such as Hindi, Kannada, Malayalam, Tamil, and Telugu accents for evaluation of the ASR system.

Figure 8.2 shows phonetic confusion matrices (obtained from pre-trained DeepSpeech2 model as in [158]) of non-native English speakers representing five distinct L1 accents, namely Hindi, Kannada, Malayalam, Tamil, and Telugu. Upon examination of the figure, it becomes apparent that variations exist in the phonetic confusions among different L1 accents when it comes to L2 speech. Upon analyzing the confusion matrices, we noticed that despite Indian English being treated as a single category, the errors or confidence mismatches in the confusion matrices indicate differences in pronunciation. Additionally, when identifying the top 10 confused phones, it is noteworthy that /m/, /f/, and /a/ consistently appear among them. A closer examination reveals that the phones they are confused with vary significantly across all the accents.

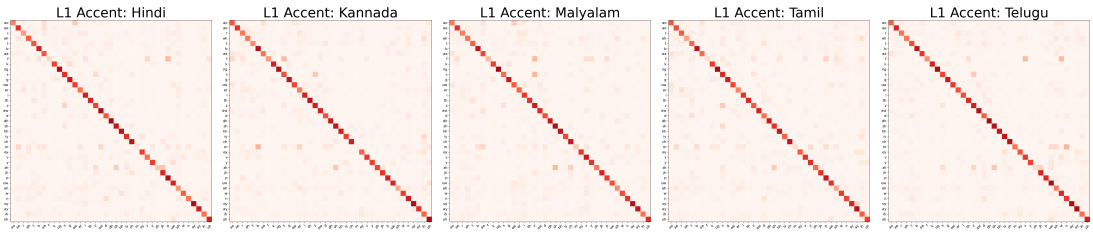


Figure 8.2: Phonetic confusion matrices (obtained from pre-trained DeepSpeech2 model) of non-native English belonging to five different L1 accents, such as Hindi, Kannada, Malayalam, Tamil, and Telugu.

Based on the observations from previous chapters for dialect classification, it can be observed that SFF-based features with ECAPA-TDNN are helpful in discriminating dialects better. To assess the relevance of these embeddings, first, we explored the ECAPA-TDNN system (along with popularly used i-vector system) for L1 identification from L2 speech. Inspired by the performance of L1 identification, i-vectors

and ECAPA-TDNN based embeddings are used to improve the performance of ASR for L2 (Indian English) speech.

Major contributions of this section are as follows:

1. Proposal of ECAPA-TDNN system for L1 identification from L2 (Indian English) speech.

2. Effectiveness of L1 embeddings (i-vectors and ECAPA-TDNN embeddings) for improving the performance of ASR for L2 speech.

3. Investigation of combined L1 embeddings for improving the performance of ASR for L2 speech.

8.3.1 Multi-lingual multi-accent corpus

This study uses the NITK-IISc Multi-lingual Multi-accent Speaker Profiling (NISP) corpus [3], which was collected to develop automatic identification of physical characteristics (such as age, height, weight, and accent). The corpus includes the non-native English (L2) speech data of native speakers of five different Indian languages (L1) such as Hindi, Kannada, Malayalam, Tamil, and Telugu.

NISP corpus is recorded with a high quality microphone i.e., "Scarlett solo studio, CM25 a large diaphragm condenser". The sampling frequency of the recorded data is 44.1 kHz, and it is re-sampled to 16 kHz.

It is a read speech collected along with the speaker's physical parameters as well as regional information and linguistic information. It is collected from faculty, students, and academia. The regional distribution of the NISP corpus is given in Figure 8.3. The English of Telugu, Tamil, Malayalam, and Kerala speakers is collected from Andhra Pradesh, Tamil Nadu, Kerala, and Karnataka respectively. While Hindi is collected from multiple states of North India.

The text used in reading for corpus contains 2 common sentences taken from TIMIT, 3 common sentences from news articles, 20 - 25 unique sentences without context from daily news articles, and 20 - 25 unique sentences with context from short stories. This approximates 45 - 55 sentences from each speaker.

This corpus, in total, contains 38.23 hrs of speech, but only 36.88 hrs of speech that has transcription is considered. The speakers are divided into training (60%), validation (20%), and test (20%) sets such that a similar distribution is observed in gender. Figure 8.4 shows the distribution of speakers across train, validation and test sets. The left bar graph shows the distribution of female speakers, and the right bar graph shows the distribution of male speakers for each L1 class across datasets. The speakers are divided such that 60% of them are in the training set, with 40% divided between dev and test sets.

This resulted in the distribution of utterances as in Table 8.2. From the total of 13353 utterances, as a result of speaker distribution (with respect to gender) led to 8538 utterances in training, 2424 utterances in

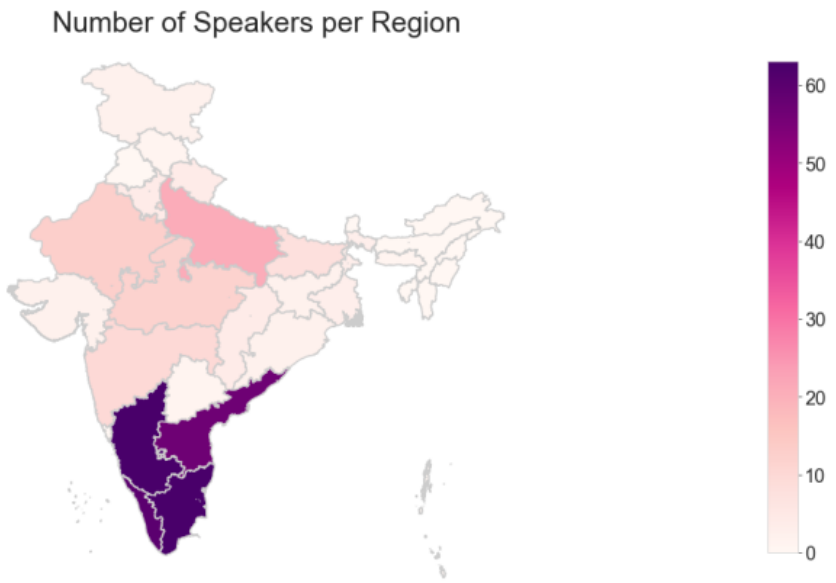


Figure 8.3: Regional distribution of NISP corpus [3]

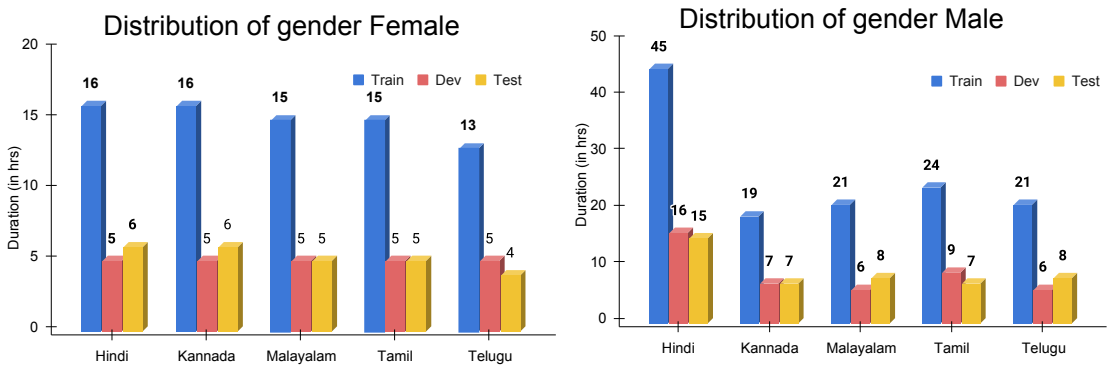


Figure 8.4: Distribution of speakers (Female on left and Male on right) across train, dev, and test sets with respect to L1 classes for NISP corpus .

validation, and 2391 utterances in test sets. The number of utterances across each class is evenly distributed, with slightly higher strength for the Hindi class.

The distribution based on gender resulted in the distribution of speech in duration across train, validation, and test as in Figure 8.5.

More details about the NISP corpus are in [3]. As per our knowledge, this is the only publicly available corpus that contains non-native English (L2) speech of native speakers of five different Indian languages (L1), and it can be downloaded from [159].

Table 8.2: Number of utterances in training, validation, and test sets of NISP corpus with respect to all five L1 accents (Hindi, Kannada, Malayalam, Tamil, and Telugu).

	No. of utterances			
L1	**Training**	**Validation**	**Test**	**Overall**
Hindi	2359	776	624	3759
Kannada	1609	512	464	2585
Malayalam	1531	321	469	2321
Tamil	1491	477	349	2317
Telugu	1548	338	485	2371
Total	**8538**	**2424**	**2391**	**13,353**

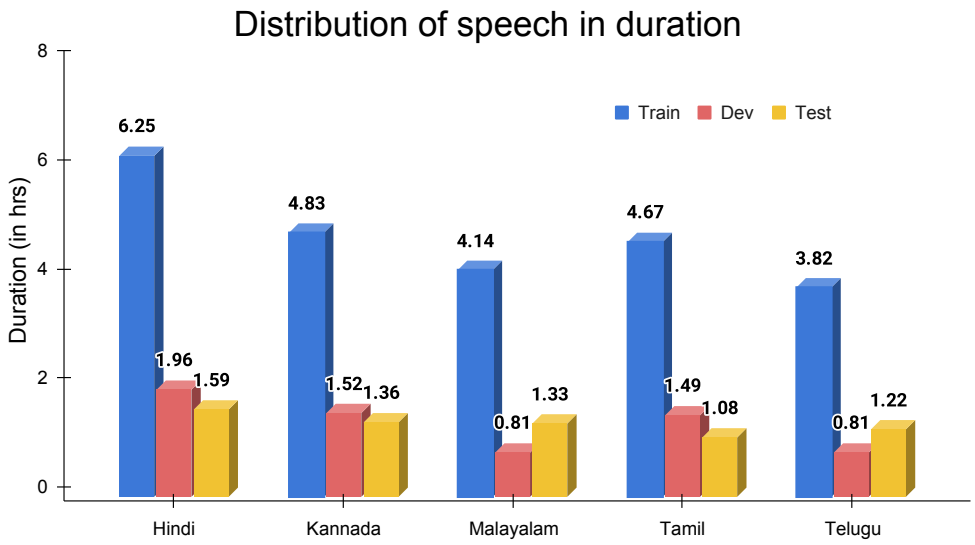


Figure 8.5: Distribution of speech across train, dev, and test sets with respect to L1 classes in duration for NISP corpus.

After carefully listening to each L1 speaker for the word "parental", it is observed that the /e/ is varied for /e/ to /æ/. It is also observed that the fluency of English spoken has different levels in each class. Changes in intonation are also observed, while there are no variations in vocabulary.

8.3.2 L1 identification from L2 speech

From the previous chapters, it can be observed that the traditional i-vector system can be used to capture all the other speaker variations along with dialects. However, ECAPA-TDNN performed better than an i-vector system for dialect classification. So, this section provides details and performances of (i-vector system and ECAPA-TDNN system) from L2 (English) speech.

8.3.2.1 Feature extraction

From previous chapters that investigated different feature representations, it is observed that among proposed MFCC-SFF performed well both with i-vector and DNNs. Along with best-performing features, the baseline features MFCC-STFT are used for L1 identification. This section gives an overview of these feature extraction methods.

MFCC-STFT: The extraction process of MFCC-STFT features is given in Chapter 4. MFCC-STFT are the cepstral coefficients extracted from log mel STFT spectrogram. 13 and 80-dimensional MFCC (extracted with 25 msec Hamming window with half of it as a shift) are used as input in i-vector system and ECAPA-TDNN system, respectively.

MFCC-SFF: The MFCC-SFF features are extracted as given in Chapter 5. From the MFCC-SFF spectrogram, mel filter bank energies (MFBE) are obtained using 80 mel filter banks. DCT is applied over a log of MFBE-SFF to obtain MFCC-SFF. For the i-vector system, the first 20 cepstral coefficients of MFCC-SFFs are used. While with the ECAPA-TDNN system, 80-dimensional MFCC-SFFs are used.

8.3.2.2 i-vector system

For dialect/accent identification, an unsupervised approach to extract i-vectors [64] from MFCC-STFTs are traditionally used [21, 22, 29, 105, 160]. Support vector machine (SVM) is used to identify the L1 of the speaker from i-vectors, and it is referred to as 'i-vector system'. A 100-dimensional i-vector is considered for this study. The i-vectors derived from an utterance retain the factors that are unique across the utterance. These factors include not only accentual features but also other speaker characteristics.

8.3.2.3 ECAPA-TDNN system

The embeddings derived from supervised neural networks tend to contain more dialect/accent information [125, 133, 161]. From our previous investigations on dialect classification in [161], Emphasized channel attention, propagation and aggregation in TDNN (ECAPA-TDNN) system [137] outperformed all the other neural networks. Motivated by this, the ECAPA-TDNN system is used for L1 identification from

L2 speech. The architecture of the ECAPA-TDNN system is the same as in [137] (except the output of the final fully-connected layer, which is set to 100).

8.3.2.4 Results of L1 identification

Two systems, i-vector and ECAPA-TDNN systems, were investigated for L1 identification. These systems were investigated with baseline (MFCC-STFT) and proposed (MFCC-SFF) features.

i-vector system: Table 8.3 shows the performance (in accuracy, unweighted average recall (UAR), and class-wise accuracies) of the i-vector system with baseline MFCC-STFT and proposed MFCC-SFF for L1 identification from L2 (English) speech. It can be observed that both the systems performed better than chance-level accuracy (20%) for all the classes. The i-vector system with MFCC-STFT features performed with 75.88% accuracy and 74.21% UAR. The class-wise accuracies show that the accuracies of all the classes are greater than 70% except for the Malayalam class. On comparison of MFCC-SFF with MFCC-STFT, it can be observed that the proposed MFCC-SFF performed better than MFCC-STFT by 3.11% relative in accuracy. It can also be observed that there is a slight improvement in the performance of the Malayalam class.

Table 8.3: Performance (in accuracy (ACC.) and unweighted average recall (UAR)) of i-vector system for L1 identification from L2 speech. Class-wise accuracies are also reported.

Features	**ACC.**	**UAR**	**Hindi**	**Kannada**	**Malayalam**	**Tamil**	**Telugu**
MFCC-STFT	**75.88**	**74.21**	90.01	78.88	40.72	74.50	86.91
MFCC-SFF	**78.24**	**77.32**	86.69	78.66	44.99	83.09	93.17

Figure 8.6 shows the confusion matrices i-vector based L1 identification system with MFCC-STFT and MFCC-SFF features. From baseline MFCC-STFT features, it can be observed that the Malayalam class is highly confused with Tamil and Hindi. This confusion in the Malayalam class is reduced with MFCC-SFF features, and further, an increase in the class accuracies of other classes is observed too.

ECAPA-TDNN model: Table 8.4 shows the performance (in ACC., UAR, and class-wise accuracies) of ECAPA-TDNN system with baseline MFCC-STFT and proposed MFCC-SFF for L1 identification from L2 (English) speech. On comparison of performances of ACC. and UAR of ECAPA-TDNN to i-vectors system, it can be observed that both MFCC-STFT and MFCC-SFF have shown an improvement. MFCC-STFT has shown an improvement of 2.82% ACC. at the same time, MFCC-SFF has shown an improvement of 3.07 % in comparison to i-vectors. Comparing baseline and proposed features with ECAPA-TDNN for L1 identification, it can be observed that proposed MFCC-SFF performed better than MFCC-STFT by 3.36% ACC. (relative). All the class-wise accuracies are greater than 60% for MFCC-SFF features, showing

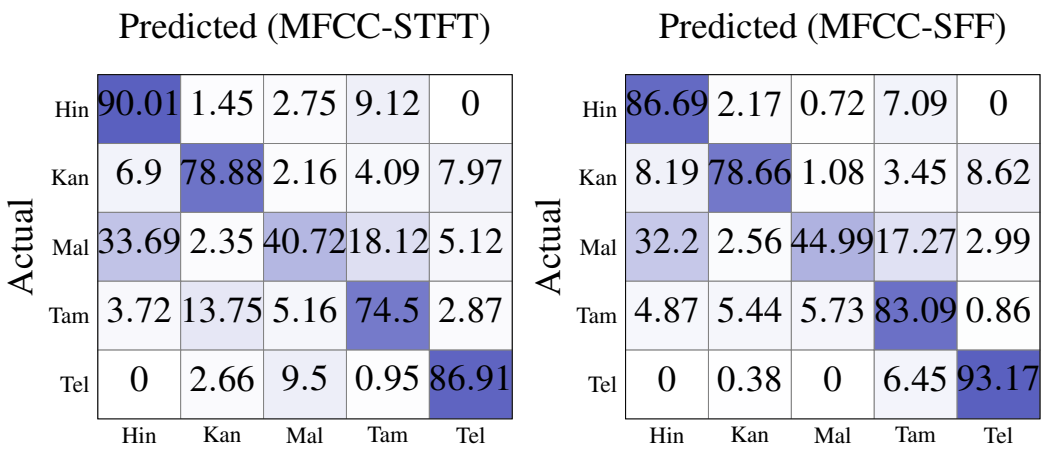


Figure 8.6: Confusion Matrices for L1 identification system with i-vector system for both baseline (MFCC-STFT) and proposed (MFCC-SFF) features.

an improvement for Malayalam class. Overall, MFCC-SFF with ECAPA-TDNN performed with 80.64% accuracy. It can be recommended to use MFCC-SFF, which provides high spectral resolution for closer dialects.

Table 8.4: Performance (in accuracy (ACC.) and unweighted average recall (UAR)) of ECAPA-TDNN system for L1 identification from L2 speech. Class-wise accuracies are also reported.

Features	**ACC.**	**UAR**	**Hindi**	**Kannada**	**Malayalam**	**Tamil**	**Telugu**
MFCC-STFT	**78.02**	**77.71**	75.69	94.11	51.92	74.07	92.76
MFCC-SFF	**80.64**	**79.27**	76.77	93.88	61.83	71.18	92.69

It is to be noted that the proposed results are better than the recent previous study [162]. However, the results are not comparable as the amount of data in [162] considers only 55 speakers instead of all speakers as in the current study.

Figure 8.7 shows the confusion matrices ECAPA-TDNN based L1 identification system with MFCC-STFT and MFCC-SFF features. From baseline MFCC-STFT features, it can be observed that the Malayalam class is highly confused with Tamil and Hindi. A confusion of Hindi to Tamil and Tamil to Hindi is also observed. Among all the classes, Telugu seemed to be a less confusing and well-classified class. On comparison of ECAPA-TDNN to i-vector based confusion matrices (in Figure 8.6), the confusion of the Malayalam class has reduced. The confusion in the Malayalam class is reduced with MFCC-SFF trained ECAPA-TDNN based L1 identification system.

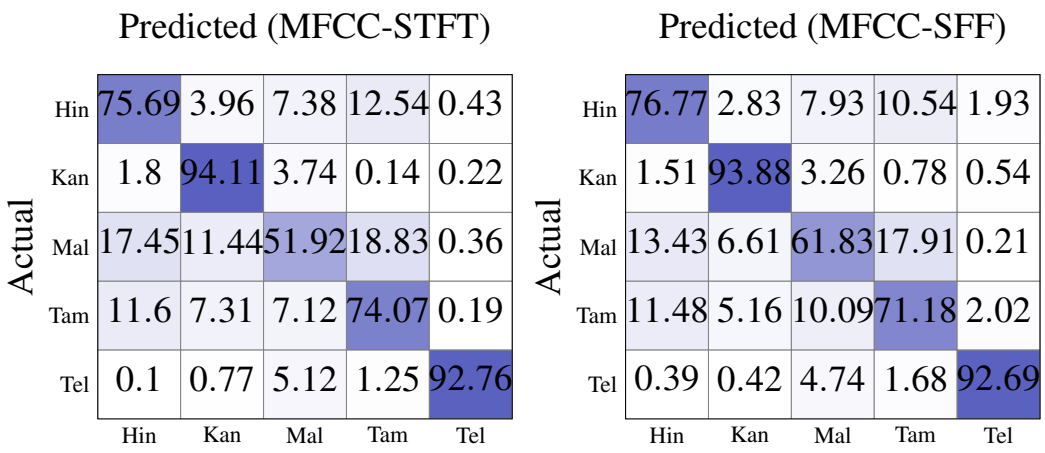


Figure 8.7: Confusion Matrices for L1 identification system with ECAPA-TDNN model for both baseline (MFCC-STFT) and proposed (MFCC-SFF) features.

Inspired by the results, ECAPA-TDNN embeddings (size of 100) are extracted from the final fully-connected layer (after the FC layer in Figure 7.9) from model trained with both MFCC-STFT and MFCC-SFF. With the effectiveness of L1 identification using i-vector system and ECAPA-TDNN system, i-vectors and ECAPA-TDNN embeddings are considered, and they are referred to as 'L1 embeddings'. The L1 embeddings derived from the model trained with MFCC-STFT are called 'STFT L1 embeddings' and with MFCC-SFF are called 'SFF L1 embeddings'. For consistency, the sizes of both L1 embeddings (i-vectors and ECAPA-TDNN) are considered 100. Both of these L1 embeddings are explored with DeepSpeech2 based ASR system for improving the performance of ASR with Indian English (L2 speech) with five different L1 accents.

8.3.3 Leveraging L1 embeddings in Indian English ASR system

This section initially reports the performance of the pre-trained DeepSpeech2 model (see Section 8.1.0.1) and then reports the performance of fine-tuned DeepSpeech2 model (see Section 8.1.0.2). Later, the effectiveness of the L1 embeddings (see Section 8.1.0.3) with DeepSpeech2 is discussed. Finally, combined L1 embeddings with the DeepSpeech2 model (see Section 8.1.0.4) is discussed. Table 8.5 reports the performance in WER% for four variants of the ASR model (along rows) with respect to five L1 accents (Hindi, Kannada, Malayalam, Tamil, and Telugu) (columns 2-6) of Indian English. The table also reports average WER (column 7), relative improvement with respect to the pre-trained DeepSpeech2 model (denoted as 'Rel. imp. w.r.t. pre-trained') (column 8), and relative improvement with respect to fine-tuned DeepSpeech2 model (denoted as 'Rel. imp. w.r.t. fine-tuned') (column 9).

The pre-trained DeepSpeech2 model is fine-tuned with 23.71 hrs (training data) of NISP corpus that includes a speech from all five L1 accents. The pre-trained DeepSpeech2 is trained for 100 epochs to obtain fine-tuned DeepSpeech2 model. The results and discussion corresponding to this system are given in Section 8.3.3.2.

Table 8.5: Performance (in WER%) of ASR systems (pre-trained, fine-tuned, i-vector based L1 embeddings, ECAPA-TDNN based L1 embeddings, and combined L1 embeddings) for five different L1 accents of Indian English. Rel. imp. refers to relative improvement.

Model Type	**Hindi**	**Kannada**	**Malayalam**	**Tamil**	**Telugu**	**Average**	**Rel. imp. w.r.t. pre-trained**	**Rel. imp. w.r.t. fine-tuned**
Pre-trained DeepSpeech2								
Pre-trained DeepSpeech2	54.40	39.26	51.77	65.86	53.27	52.91	-	-
Fine-tuned DeepSpeech2								
Fine-tuned DeepSpeech2	50.33	31.01	49.29	57.22	44.43	46.46	12.20	-
L1 class with Fine-tuned DeepSpeech2								
One-hot encoded L1 classes	38.52	21.15	35.56	43.25	29.30	33.56	36.57	27.77
STFT L1 embeddings with Fine-tuned DeepSpeech2								
i-vector based L1 emb.	40.02	21.73	37.83	45.43	31.56	35.32	33.26	23.98
ECAPA-TDNN based L1 emb.	41.75	23.09	38.50	43.70	33.46	36.10	31.78	22.29
SFF L1 embeddings with Fine-tuned DeepSpeech2								
i-vector based L1 emb.	41.13	21.54	36.78	43.87	29.59	34.58	34.64	25.43
ECAPA-TDNN based L1 emb.	40.72	19.04	34.00	40.72	27.80	32.45	38.66	30.14
Combined L1 embeddings with DeepSpeech2								
STFT i-vector+ECAPA-TDNN emb.	39.57	21.70	36.10	43.847	29.88	34.22	34.64	26.34
SFF i-vector+ECAPA-TDNN emb.	**32.92**	**15.77**	**28.31**	**34.70**	**21.66**	26.67	49.60	42.59

8.3.3.1 Pre-trained DeepSpeech2

Row 3 of Table 8.5 shows the performance (in WER%) of the pre-trained DeepSpeech2 model. From the table, it can be observed that the average performance of the pre-trained DeepSpeech2 model is 52.91% WER. Among the five L1 accents, Kannada performed better than all the other L1 accents. It can also be observed that Kannada and Malayalam accented English are above the average, and Hindi, Tamil, and Telugu accented English are below the average performance.

8.3.3.2 Impact of fine-tuned DeepSpeech2

Row 5 of Table 8.5 shows the performance of fine-tuned DeepSpeech2 model. It can be observed that fine-tuned DeepSpeech2 model gave a relative improvement of 20.58% WER over the pre-trained DeepSpeech2 model. In comparison to the pre-trained DeepSpeech2 model, a consistent improvement can be observed for all five L1 accents with an absolute improvement of 8.45%, 11.67%, 6.29%, 12.49%, and 14.72% for Hindi, Kannada, Malayalam, Tamil, and Telugu, respectively. This is expected as the fine-tuned network learns the overall variations of L1 accents.

8.3.3.3 Proof of concept with ground truth L1 class

Row 7 of Table 8.5 shows the performance of a one-hot encoded L1 class with fine-tuned DeepSpeech2 model. It can be observed that providing the L1 class with fine-tuned DeepSpeech2 model gave a relative improvement of 36.57% WER over the pre-trained DeepSpeech2 model. In comparison to fine-tuned DeepSpeech2 model, providing L1 information improved the performance of fine-tuned DeepSpeech2 models by 27.77% WER (relative) on an average for all L1 classes. This shows that providing the native (L1) language information to the Indian English ASR model with speakers from different L1s will improve speech recognition performance.

It can also be observed that there is a consistent improvement for all five L1 accents with a relative improvement of 29.19%, 46.13%, 32.61%, 34.33%, and 45.00% in WER for Hindi, Kannada, Malayalam, Tamil, and Telugu, respectively.

8.3.3.4 Impact of L1 embeddings with DeepSpeech2

Rows 8-13 of Table 8.5 show the DeepSpeech2 model with i-vector based L1 embeddings and ECAPA-TDNN based L1 embeddings, respectively. It can be clearly seen that the inclusion of L1 embeddings in the DeepSpeech2 model significantly improved the performance when compared to pre-trained and fine-tuned DeepSpeech2 models. This indicates that adding the information about the L1 accent of the speaker improves the performance of the ASR system for L2 speech. It can also be observed that continuous latent representations i.e., L1 embeddings have shown similar performance when compared to ground truth L1 classes.

The i-vector L1 embeddings derived using MFCC-STFT features gave a relative improvement of 33.26% and 23.98% (in WER) over pre-trained and fine-tuned DeepSpeech2 models, respectively. While, i-vector L1 embeddings derived using proposed MFCC-SFF features gave a relative improvement of 34.64% and 25.43% (in WER) over pre-trained and fine-tuned DeepSpeech2 models, respectively. On comparison of STFT L1 embeddings and SFF L1 embeddings of the i-vector system, it can be observed that proposed

SFF-based features outperformed. This shows that the SFF embeddings not only improved dialect classification but also the embeddings derived from SFF are better in speech recognition.

With STFT based ECAPA-TDNN L1 embeddings, fine-tuned DeepSpeech2 model (refer to row 10 of Table 8.5) gave an improvement of 31.78% and 22.29% over pre-trained and fine-tuned DeepSpeech2 models, respectively. While with SFF based ECAPA-TDNN L1 embeddings, fine-tuned DeepSpeech2 model (refer to row 13 of Table 8.5) gave an improvement of 38.66% and 30.14% over pre-trained and fine-tuned DeepSpeech2 models, respectively.

On comparison of i-vector L1 embeddings to ECAPA-TDNN based L1 embeddings with STFT features (see rows 9 and 10 of Table 8.5), it can be observed that only ECAPA-TDNN performed better for only Tamil while other performed reasonably well. On comparison of i-vector L1 embeddings to ECAPA-TDNN based L1 embeddings with SFF features (see rows 12 and 13 of Table 8.5), it can be observed that only ECAPA-TDNN performed better for all the L1 classes with an overall improvement of 6.16% WER (relative).

Overall between both i-vectors and ECAPA-TDNN, both seemed to be important. Both combined might improve speech recognition performance further.

8.3.3.5 Impact of combined i-vectors and ECAPA-TDNN L1 embeddings with DeepSpeech2

The last two rows of Table 8.5 show the results of combined L1 embeddings with the DeepSpeech2 model. From the table, it can be observed that combined L1 embeddings derived from SFF with the DeepSpeech2 model outperformed all the other variants. Both the STFT combined (i-vectors and ECAPA-TDNN) L1 embeddings and SFF combined L1 embeddings have shown an improvement when compared to individual embeddings. This indicates that there exists complementary information between i-vector based L1 embeddings and ECAPA-TDNN based L1 embeddings. Combined STFT L1 embeddings with the DeepSpeech2 model gave a relative improvement of 34.64% and 26.34% with respect to pre-trained and fine-tuned DeepSpeech2 models, respectively. While combined SFF L1 embeddings with the DeepSpeech2 model gave a relative improvement of 49.60% and 42.59% with respect to pre-trained and fine-tuned DeepSpeech2 models, respectively.

Also, it can be seen that combined L1 embeddings with the DeepSpeech2 model performed better for almost all the L1 accents compared to i-vector based L1 embeddings and ECAPA-TDNN based L1 embeddings, indicating the existence of complementary information in both L1 embeddings. It is also observed that SFF combined L1 embeddings performed better than one-hot encoded ground L1 class showing that the L1 embeddings derived from the proposed SFF approach better represented accentual traits in speech.

8.4 Summary and conclusions

To observe how the improved dialect embeddings can help speech recognition, the embeddings derived from the proposed approach (i-vectors and ECAPA-TDNN embeddings from MFCC-SFF) are investigated with multi-dialect speech recognition. From the experimentation, it is observed that the dialectal embeddings that contain dialectal traits are advantageous for multi-dialect speech recognition. Further, with SFF based i-vectors and ECAPA-TDNN embeddings which were better in the classification of dialects are more advantageous in multi-dialect ASR system.

In this chapter, we proposed to use L1 embeddings (i-vectors and embeddings extracted from ECAPA-TDNN) for improving the ASR performance of Indian English (L2). The relevance of these embeddings was assessed by developing L1 identification systems. Five variants of DeepSpeech2 ASR models were developed for L2 speech. They are: pre-trained, fine-tuned, i-vector based embedding, ECAPA-TDNN based embedding, and combined embedding models. With STFT features, the i-vector based embeddings and ECAPA-TDNN based embeddings gave better performance compared to the pre-trained model by 33.26% and 31.78% and compared to fine-tuned model by 23.98% and 22.29% in WER (relative), respectively. With SFF features, the i-vector based embeddings and ECAPA-TDNN based embeddings gave better performance compared to the pre-trained model by 34.64% and 38.66% and compared to fine-tuned model by 25.43% and 30.14% in WER (relative), respectively. Between the two features, it is observed that SFF based (MFCC-SFF) helped to better represent L1 information that is useful for speech recognition. Between the two embeddings, both seem to have L1 information and i-vectors also have other variabilities (such as speaker, environment, and so on) that can improve ASR performance. So, combined embeddings (i-vectors and ECAPA-TNN embeddings) are leveraged to use complementary information. It is observed that there exists complementary information. Further, it can be concluded that with SFF based combined L1 embeddings helped to reduce the word error of the fine-tuned DeepSpeech2 model from 46.46% WER to 26.67% WER which gave a 42.59% improvement relatively.

Chapter 9

Summary and Conclusions

Due to multi-dialectal speech in real-time scenarios, the performance of the automatic speech recognization (ASR) system will be degraded. It is observed from previous studies that the performance of the acoustic model of speech recognizer can be improved with accent information. Previously, the dialect embeddings or continuous representations derived from dialect classifiers were extracted using traditional dialect classification approaches. The improved dialect embeddings that can represent dialects better can improve the performance of the ASR system. This thesis is presented in three steps.

As a first step, different feature extraction methods were proposed for dialect classification. Dialectal variations can be observed due to dynamic or transient sounds such as trill and aspiration, which are not well represented by current STFT features. So, this thesis proposed to use features derived from signal processing approaches, such as single frequency filtering (SFF) and zero-time windowing(ZTW) methods. The features derived from these methods provide higher temporal resolution without compromising spectral resolution. ZTWCC and MFCC-ZTW features derived using the ZTW method performed better than MFCC-STFT features with the traditional i-vector dialect classification system. Also, SFFCC and MFCC-SFF derived using the SFF method performed better than baseline MFCC-STFT features. s

Further, dialectal variations can also be observed as longer temporal variations in speech. FDLPCC features extracted from the frequency domain linear prediction are investigated for dialect classification to capture this longer temporal summarization. The FDLPCC features with the i-vector system performed better than baseline MFCC-STFT features. From experimentation, it is concluded that in the i-vector system, MFCC-SFF and FDLPCC features performed with a UAR of 81.25% and 81.37% UAR, respectively.

As a second step, deep neural networks were proposed for dialect classification. In traditional i-vector-based approaches, the delta coefficients include temporal context in each frame. It is observed that a longer temporal context is advantageous for dialect classification. So, the convolution neural network (CNN) that provides a more extended temporal context with non-linear computations is investigated for dialect classification. Different variants of CNN, such as TCNN, TDNN, and ECAPA-TDNN, are investigated with proposed features to provide different temporal contexts. From the experiments, it is observed that

SFF-based features performed well with longer temporal context (SDCs) in traditional i-vector approach showed a similar improvement with better deep neural network architecture—the MFCC-SFF derived from SFF method performed with a UAR of 83.22% for dialect classification. The longer temporal context by the neural network is advantageous with MFCC-SFF for dialect classification.

As a third step, the embeddings derived from improved dialect classification are leveraged in a multi-dialectal ASR system. The proposed MFCC-SFF features that outperformed all the other features with i-vector and ECAPA-TDNN systems are leveraged in multi-dialect speech recognition systems. Out of baseline MFCC-STFT embeddings and MFCC-SFF embeddings leveraged in multi-dialect ASR, MFCC-SFF embeddings have shown slight improvement.

In most speech recognition models, Indian English is considered one class, even though it has speakers from many native (L1) languages. So, this thesis proposed to use L1 embeddings in the Indian English ASR system to improve its performance. Based on the observations till now, MFCC-SFF features with both i-vector and ECAPA-TDNN were observed to be better for dialect classification. So, we propose to use the i-vectors and ECAPA-TDNN embeddings derived using MFCC-SFF for L1 identification. Further, we propose to use the i-vectors and ECAPA-TDNN (L1) embeddings in the Indian English ASR system. Likely to dialect classification, proposed MFCC-SFF features performed better for L1 identification. It is also observed that the L1 embeddings derived using MFCC-SFF are more beneficial for the Indian ASR system. SFF-based combined L1 embeddings improved the performance of the fine-tuned DeepSpeech2 model from 46.46% WER (finetuned model) to 26.67% WER, which is a relative improvement of 42.59% WER.

This thesis worked on dialects of English. In the future, similar studies can be extended to dialects of other languages. One of our studies also observed that the proposed SFF and ZTW features performed better for classifying dialects in German.

The embeddings derived using MFCC-SFF features with ECAPA-TDNN can be used for Indian languages to improve the acoustic models of multi-dialect Indian languages. Since the languages are agglutinative and different words are formed based on dialects, approaches to handling language models should also be in place. The word embeddings learned in neural language models are clustered based on semantics, but it is blind to the internal structure of each word. Especially in agglutinative Indian languages, a vast vocabulary of synonyms can be formed from a stem word with different affixes across dialects. The word's internal structure will help cluster the synonyms in such cases. In such languages using the sub-word information such as characters, morphemes, or syllables, along with word embeddings, improved the performance of language models [163–167]. From [168], it can be observed that syllable-level representations are better than a character in morphologically rich languages. Inspired by that, we propose investigating the importance of syllable information for word-level language models for dialectal Telugu datasets. Based on this motivation, Telugu speech corpus can be collected for different dialects. Then, along

with the proposed acoustic model, syllable-aware word-level language can be incorporated as part of the dialectal Telugu ASR system.

Further, these studies can be extended to language identification and the inclusion of language embeddings for a unified ASR system for all languages.

Bibliography

[1] H. Ding, O. Jokisch, and R. Hoffmann, "F0 analysis of Chinese accented German speech," in *Proc. ISCSLP*, 2006, pp. 49–56.

[2] A. Levent and J. H. Hansen, "A study of temporal features and frequency characteristics in American English foreign accent," *The Journal of the Acoustical Society of America (JASA)*, vol. 102, no. 1, pp. 28–40, 1997.

[3] S. B. Kalluri, D. Vijayasenan, S. Ganapathy, R. R. M, and P. Krishnan, "NISP: A multi-lingual multi-accent dataset for speaker profiling," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2021, pp. 6953–6957.

[4] C. Huang, T. Chen, S. Z. Li, E. Chang, and J.-L. Zhou, "Analysis of speaker variability." in *Proc. Interspeech*, 2001, pp. 1377–1380.

[5] J. C. Wells, *Accents of English*. Cambridge University Press, 1982, vol. 1.

[6] S. Weinberger and S. Kunath, "The speech accent archive: Towards a typology of English accents," *Language and Computers*, vol. 73, 2011.

[7] P. Gomez, "British and american english pronunciation differences," 2009.

[8] C. Huang, T. Chen, and E. Chang, "Accent issues in large vocabulary continuous speech recognition," *Int. Journal of Speech Technology*, vol. 7, no. 2-3, pp. 141–153, 2004.

[9] A. Jain, M. Upreti, and P. Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning," in *Proc. Interspeech*, 2018, pp. 2454–2458.

[10] M. T. Turan, E. Vincent, and D. Jouvet, "Achieving multi-accent ASR via unsupervised acoustic model adaptation," in *Proc. Interspeech*, 2020, pp. 1286–1290.

[11] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel, "Racial disparities in automated speech recognition," *National Academy of Sciences*, vol. 117, no. 14, pp. 7684–7689, 2020.

[12] X. Yang, K. Audhkhasi, A. Rosenberg, S. Thomas, B. Ramabhadran, and M. Hasegawa-Johnson, "Joint modeling of accents and acoustics for multi-accent speech recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2018, pp. 1–5.

[13] D. Vergyri, L. Lamel, and J.-L. Gauvain, "Automatic speech recognition of multiple accented English data," in *Proc. Interspeech*, 2010, pp. 1652–1655.

[14] K. Kulkarni, S. Sengupta, V. Ramasubramanian, J. G. Bauer, and G. Stemmer, "Accented Indian English ASR: Some early results," in *Proc. Spoken Language Technology Workshop*, 2008, pp. 225–228.

[15] Y. Zheng, R. Sproat, L. Gu, I. Shafran, H. Zhou, Y. Su, D. Jurafsky, R. Starr, and S.-Y. Yoon, "Accent detection and speech recognition for Shanghai-accented Mandarin," in *Proc. Interspeech*, 2005, pp. 217–220.

[16] N. T. Vu, Y. Wang, M. Klose, Z. Mihaylova, and T. Schultz, "Improving ASR performance on non-native speech using multilingual and crosslingual information," in *Proc. Interspeech*, 2014, pp. 11–15.

[17] U. Nallasamy, "Adaptation techniques to improve asr performance on accented speakers," Ph.D. dissertation, Carnegie Mellon University, 2016.

[18] G. Peng and W. S.-Y. Wang, "An innovative prosody modeling method for Chinese speech recognition," *Int. Journal of Speech Technology*, vol. 7, no. 2-3, pp. 129–140, 2004.

[19] K. Almeman and M. Lee, "Automatic building of Arabic multi dialect text corpora by bootstrapping dialect words," in *Proc. Int. Conference on Communications, Signal Processing, and their Applications*, 2013, pp. 1–6.

[20] X. Huang, J. Baker, and R. Reddy, "A historical perspective of speech recognition," *Communications of the ACM*, vol. 57, pp. 94–103, 01 2014.

[21] H. Behravan, V. Hautamäki, S. M. Siniscalchi, T. Kinnunen, and C. Lee, "I-vector modeling of speech attributes for automatic foreign accent recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 29–41, 2016.

[22] A. Hanani, M. J. Russell, and M. J. Carey, "Human and computer recognition of regional accents and ethnic groups from British English speech," *Computer Speech & Language*, vol. 27, no. 1, pp. 59–74, 2013.

[23] J. H. Hansen, U. H. Yapanel, R. Huang, and A. Ikeno, "Dialect analysis and modeling for automatic classification," in *Proc. Interspeech*, 2004, pp. 1569–1572.

[24] F. Biadsy and J. Hirschberg, "Using prosody and phonotactics in Arabic dialect identification," in *Proc. Tenth Annual Conference of the Int. Speech Communication Association*, 2009.

[25] M. Najafian, S. Safavi, P. Weber, and M. J. Russell, "Identification of British English regional accents using fusion of i-vector and multi-accent phonotactic systems," in *Proc. ODYSSEY*, 2016, pp. 132–139.

[26] F. Biadsy, J. Hirschberg, and N. Habash, "Spoken Arabic dialect identification using phonotactic modeling," in *Proc. Workshop on Computational Approaches to Semitic Languages*, 2009, pp. 53–61.

[27] M. A. Zissman, T. P. Gleason, D. Rekart, and B. L. Losiewicz, "Automatic dialect identification of extemporaneous conversational, Latin American Spanish speech," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 1996, pp. 777–780 vol. 2.

[28] F. S. Richardson, W. M. Campbell, and P. A. Torres-Carrasquillo, "Discriminative n-gram selection for dialect recognition," in *Proc. Interspeech*, 2009, pp. 192–195.

[29] H. Behravan, V. Hautamäki, and T. Kinnunen, "Foreign accent detection from spoken Finnish using i-vectors," in *Proc. Interspeech*, 2013, pp. 79–83.

[30] A. DeMarco and S. J. Cox, "Iterative classification of regional British accents in i-vector space," in *Proc. Symposium on Machine Learning in Speech and Language Processing*, 2012, pp. 1–4.

[31] Q. Zhang and J. H. Hansen, "Dialect recognition based on unsupervised bottleneck features." in *Proc. Interspeech*, 2017, pp. 2576–2580.

[32] ——, "Language/dialect recognition based on unsupervised deep learning," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 26, no. 5, pp. 873–882, 2018.

[33] B. W. Schuller, A. Batliner, C. Bergler, F. B. Pokorny, J. Krajewski, M. Cychosz, R. Vollmann, S.-D. Roelen, S. Schnieder, E. Bergelson10 *et al.*, "The Interspeech 2019 computational paralinguistics challenge: Styrian dialects, continuous sleepiness, baby sounds & orca activity," in *Proc. Interspeech*, 2019.

[34] K. Kumpf and R. W. King, "Foreign speaker accent classification using phoneme-dependent accent discrimination models and comparisons with human perception benchmarks," in *Proc. Eurospeech*, 1997.

[35] P. Angkititrakul and J. H. Hansen, "Advances in phone-based modeling for automatic accent classification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 634–646, 2006.

[36] H. Behravan, V. Hautamäki, and T. Kinnunen, "Factors affecting i-vector based foreign accent recognition: A case study in spoken Finnish," *Speech Communication*, vol. 66, pp. 118–129, 2015.

[37] S. Shon, A. Ali, and J. R. Glass, "Convolutional neural networks and language embeddings for end-to-end dialect recognition," *CoRR*, vol. abs/1803.04567, 2018.

[38] R. Ubale, Y. Qian, and K. Evanini, "Exploring end-to-end attention-based neural networks for native language identification," in *Proc. Spoken Language Technology Workshop*, 2018, pp. 84–91.

[39] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoustics Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[40] L. M. Arslan and J. H. Hansen, "Frequency characteristics of foreign accented speech," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, vol. 2, 1997, pp. 1123–1126.

[41] Q. Yan and S. Vaseghi, "Analysis, modelling and synthesis of formants of British, American and Australian accents," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2003, pp. 712–715.

[42] L. W. Kat and P. Fung, "Fast accent identification and accented speech recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, vol. 1, 1999, pp. 221–224.

[43] M. A. Yusnita, M. P. Paulraj, S. Yaacob, S. A. Bakar, and A. Saidatul, "Malaysian English accents identification using LPC and formant analysis," in *Proc. Int. Conf. on Control System, Computing and Engineering*, 2011, pp. 472–476.

[44] S. Safavi, A. Hanani, M. Russell, P. Jancovic, and M. J. Carey, "Contrasting the effects of different frequency bands on speaker and accent identification," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 829–832, 2012.

[45] Kethireddy, Rashmi, S. R. Kadiri, and S. V. Gangashetty, "Learning filterbanks from raw waveform for accent classification," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, 2020, pp. 1–6.

[46] K. Pike, *The Intonation of American English*, ser. University of Michigan Publications: Linguistics. University of Michigan Press, 1945, no. 1-3.

[47] D. Abercrombie *et al.*, *Elements of general phonetics*. Edinburgh University Press Edinburgh, 1967, vol. 203.

[48] F. Ramus, M. Nespor, and J. Mehler, "Correlates of linguistic rhythm in the speech signal," *Cognition*, vol. 73, no. 3, pp. 265–292, 1999.

[49] M. Nespor, "On the rhythm parameter in phonology," *Logical issues in language acquisition*, pp. 157–175, 1990.

[50] S. Ghazali, R. Hamdi, and M. Barkat, "Speech rhythm variation in Arabic dialects," in *Proc. International Conference on Speech Prosody*, 2002.

[51] D. H. Deterding, "The measurement of rhythm: a comparison of Singapore and British English," *J. Phonetics*, vol. 29, no. 2, pp. 217–230, 2001.

[52] L. E. Ling, E. Grabe, and F. Nolan, "Quantitative characterizations of speech rhythm: Syllable-timing in Singapore English," *Language and speech*, vol. 43, no. 4, pp. 377–401, 2000.

[53] H.-L. Jian, "On the syllable timing in Taiwan English," in *Proc. International Conference on Speech Prosody*, 2004.

[54] E. Ferragne and F. Pellegrino, "Rhythm in read British English: interdialect variability," in *Proc. Interspeech*, 2004, pp. 1573–1576.

[55] D. C. Zheng, D. Dyke, F. Berryman, and C. Morgan, "A new approach to acoustic analysis of two British regional accents–Birmingham and Liverpool accents," *International Journal of Speech Technology*, vol. 15, no. 2, p. 77–85, 2012.

[56] E. Grabe, "Variation adds to prosodic typology," in *Proc. International Conference Speech Prosody*, 2002.

[57] M. Mehrabani, H. Bořil, and J. H. Hansen, "Dialect distance assessment method based on comparison of pitch pattern statistical models," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2010, pp. 5158–5161.

[58] G. Peng, "Temporal and tonal aspects of Chinese syllables: A corpus-based comparative study of Mandarin and Cantonese," *Journal of Chinese Linguistics*, vol. 34, no. 1, p. 134, 2006.

[59] L. R. Yanguas and T. F. Quatieri, "Implications of glottal source for speaker and dialect identification," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, vol. 2, 1999, pp. 813–816.

[60] S. Malmasi, E. Refaee, and M. Dras, "Arabic dialect identification using a parallel multidialectal corpus," in *Proc. Computational Linguistics*, 2016, pp. 35–53.

[61] J. H. Hansen and G. Liu, "Unsupervised accent classification for deep data fusion of accent and language information," *Speech Communication*, vol. 78, pp. 19–33, 2016.

[62] M. H. Bahari, R. Saeidi, H. Van hamme, and D. Van Leeuwen, "Accent recognition using i-vector, Gaussian mean supervector and Gaussian posterior probability supervector for spontaneous telephone speech," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2013, pp. 7344–7348.

[63] M. H. Bahari, N. Dehak, H. Van hamme, L. Burget, A. M. Ali, and J. Glass, "Non-negative factor analysis of Gaussian mixture model weight adaptation for language and dialect recognition," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 22, no. 7, pp. 1117–1129, 2014.

[64] N. Dehak, "Discriminative and generative approaches for long- and short-term speaker characteristics modeling: Application to speaker verification," Ph.D. dissertation, 2009.

[65] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[66] P. A. Torres-Carrasquillo, T. P. Gleason, and D. A. Reynolds, "Dialect identification using Gaussian mixture models," in *Proc. Speaker Odyssey Workshop*, 2004, pp. 297–300.

[67] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[68] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in neural information processing systems*, 2012, pp. 1097–1105.

[70] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[71] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[72] J. H. Hansen and G. Liu, "Unsupervised accent classification for deep data fusion of accent and language information," *Speech Communication*, vol. 78, pp. 19–33, 2016.

[73] V. Gupta and P. Mermelstein, "Effects of speaker accent on the performance of a speaker-independent, isolated-word recognizer," *The Journal of the Acoustical Society of America (JASA)*, vol. 71, pp. 1581–1587, 1982.

[74] A. Faria, "Accent classification for speech recognition," in *Proc. Int. Workshop on Machine Learning for Multimodal Interaction*, 2005, pp. 285–293.

[75] F. Biadsy, "Automatic dialect and accent recognition and its application to speech recognition," Ph.D. dissertation, Columbia University, 2011.

[76] A. Abad, E. Ribeiro, F. Kepler, R. F. Astudillo, and I. Trancoso, "Exploiting phone log-likelihood ratio features for the detection of the native language of non-native English speakers," in *Proc. Interspeech*, 2016, pp. 2413–2417.

[77] J. H. Hansen and L. M. Arslan, "Foreign accent classification using source generator based prosodic features," *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, vol. 1, pp. 836–839, 1995.

[78] L. M. Arslan and J. H. Hansen, "A study of temporal features and frequency characteristics in American English foreign accent," *The Journal of the Acoustical Society of America (JASA)*, vol. 102, no. 1, pp. 28–40, 1997.

[79] L. W. Kat and P. Fung, "Fast accent identification and accented speech recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 1999, pp. 221–224.

[80] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. D. Jr., "Approaches to language identification using Gaussian mixture models and shifted delta cepstral features," in *Proc. Int. Conf. Spoken Language Processing (INTERSPEECH)*, 2002.

[81] A. DeMarco and S. J. Cox, "Native accent classification via i-vectors and speaker compensation fusion," in *Proc. Interspeech*, 2013, pp. 1472–1476.

[82] S. Shon, W.-N. Hsu, and J. Glass, "Unsupervised representation learning of speech for dialect identification," in *Proc. Spoken Language Technology (SLT) Workshop*. IEEE, 2018, pp. 105–111.

[83] A. A. Suwon Shon and J. Glass, "Convolutional neural network and language embeddings tend-to-end dialect recognition," in *Proc. The Speaker and Language Recognition Workshop (Odyssey)*, 2018.

[84] Y. Wu, H. Mao, and Z. Yi, "Audio classification using attention-augmented convolutional neural network," *Knowledge-Based Systems*, vol. 161, pp. 90–100, 2018.

[85] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoustics Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.

[86] J. D. Markel and A. J. Gray, *Linear prediction of speech*. Springer Science & Business Media, 2013, vol. 12.

[87] J. Makhoul, "Linear prediction in automatic speech recognition," *Speech Recognition*, pp. 183–220, 1975.

[88] E. Wong and S. Sridharan, "Comparison of linear prediction cepstrum coefficients and mel-frequency cepstrum coefficients for language identification," in *Proc. Int. Symposium on Intelligent Multimedia, Video and Speech Processing. ISIMP 2001 (IEEE Cat. No. 01EX489)*. IEEE, 2001, pp. 95–98.

[89] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *The Journal of the Acoustical Society of America (JASA)*, vol. 55, no. 6, pp. 1304–1312, 1974.

[90] S. Furui, "Speaker-independent isolated word recognition based on emphasized spectral dynamics," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, vol. 11. IEEE, 1986, pp. 1991–1994.

[91] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, "A study of interspeaker variability in speaker verification," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 5, pp. 980–988, 2008.

[92] H. Lei, "Joint factor analysis (jfa) and i-vector tutorial," *ICSI. Web. 02 Oct*, 2011.

[93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[94] S. O. Sadjadi, M. Slaney, and L. P. Heck, "MSR identity toolbox v1.0: A matlab toolbox for speaker recognition research," 2013.

[95] A. Rosenberg, "Classifying skewed data: importance weighting to optimize average recall," in *Proc. Interspeech*, 2012, pp. 2242–2245.

[96] V. Pannala, G. Aneeja, S. R. Kadiri, and B. Yegnanarayana, "Robust estimation of fundamental frequency using single frequency filtering approach," in *Proc. Interspeech*, 2016, pp. 2155–2159.

[97] S. R. Kadiri and B. Yegnanarayana, "Epoch extraction from emotional speech using single frequency filtering approach," *Speech Communication*, vol. 86, pp. 52–63, 2017.

[98] B. T. Nellore, R. Prasad, S. R. Kadiri, S. V. Gangashetty, and B. Yegnanarayana, "Locating burst onsets using SFF envelope and phase information," in *Proc. Interspeech*, 2017, pp. 3023–3027.

[99] G. Aneeja, S. R. Kadiri, and B. Yegnanarayana, "Detection of glottal closure instants in degraded speech using single frequency filtering analysis," in *Proc. Interspeech*, 2018, pp. 2300–2304.

[100] J. H. L. Hansen, S. S. Gray, and W. Kim, "Automatic voice onset time detection for unvoiced stops (/p/, /t/, /k/) with application to accent classification," *Speech Communication*, vol. 52, no. 10, pp. 777–789, 2010.

[101] B. Yegnanarayana and N. Dhananjaya, "Spectro-temporal analysis of speech signals using zero-time windowing and group delay function," *Speech Communication*, vol. 55, no. 6, pp. 782–795, 2013.

[102] N. Dhananjaya, B. Yegnanarayana, and P. Bhaskararao, "Acoustic analysis of trill sounds," *The Journal of the Acoustical Society of America (JASA)*, vol. 131, no. 4, pp. 3141–3152, 2012.

[103] N. Dhananjaya, "Signal processing for excitation-based analysis of acoustic events in speech," Ph.D. dissertation, Dept. of Computer Science and Engineering, IIT Madras, Chennai, Oct. 2011. [Online]. Available: speech.iiit.ac.in/svlpubs/phdthesis/dhanu-phd-2011.pdf

[104] B. Yegnanarayana and N. Dhananjaya, "Spectro-temporal analysis of speech signals using zero-time windowing and group delay function," *Speech Communication*, vol. 55, no. 6, pp. 782–795, 2013.

[105] Kethireddy, Rashmi, S. R. Kadiri, S. Kesiraju, and S. V. Gangashetty, "Zero-time windowing cepstral coefficients for dialect classification," in *Proc. ODYSSEY*, 2020, pp. 32–38.

[106] G. Aneeja and B. Yegnanarayana, "Single frequency filtering approach for discriminating speech and nonspeech," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 705–717, 2015.

[107] Kethireddy, Rashmi, S. R. Kadiri, P. Alku, and S. V. Gangashetty, "Mel-weighted single frequency filtering spectrogram for dialect identification," *IEEE Access*, vol. 8, pp. 174 871–174 879, 2020.

[108] S. R. Kadiri and B. Yegnanarayana, "Analysis and detection of phonation modes in singing voice using excitation source features and single frequency filtering cepstral coefficients (SFFCC)," in *Proc. Interspeech*, 2018, pp. 441–445.

[109] J. Mielke, C. Carignan, and E. R. Thomas, "The articulatory dynamics of pre-velar and pre-nasal /æ/-raising in English: An ultrasound study," *The Journal of the Acoustical Society of America (JASA)*, vol. 142, no. 1, pp. 332–349, 2017.

[110] R. Fox and E. Jacewicz, "Cross-dialectal variation in formant dynamics of American English vowels," *The Journal of the Acoustical Society of America (JASA)*, vol. 126, pp. 2603–2618, 2009.

[111] Y. Huang, D. Guo, A. Kasakoff, and J. Grieve, "Understanding U.S. regional linguistic variation with twitter data analysis," *Computers, Environment and Urban Systems*, vol. 59, pp. 244 – 255, 2016.

[112] M. Athineos and D. P. W. Ellis, "Frequency-domain linear prediction for temporal features," in *Proc. Automatic Speech Recognition and Understanding (ASRU) Workshop*, 2003, pp. 261–266.

[113] M. Athineos and D. Ellis, "Sound texture modelling with linear prediction in both time and frequency domains," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, vol. 5, 2003, pp. V–648.

[114] J.-L. Rouas, "Automatic prosodic variations modeling for language and dialect discrimination," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 6, pp. 1904–1911, 2007.

[115] N. F. Chen, W. Shen, J. P. Campbell, and P. A. Torres-Carrasquillo, "Informative dialect recognition using context-dependent pronunciation modeling," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2011, pp. 4396–4399.

[116] N. F. Chen, S. W. Tam, W. Shen, and J. P. Campbell, "Characterizing phonetic transformations and acoustic differences across English dialects," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 110–124, 2014.

[117] R. Huang, J. H. L. Hansen, and P. Angkititrakul, "Dialect/accent classification using unrestricted audio," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 15, no. 2, pp. 453–464, 2007.

[118] S. Ganapathy and H. Hermansky, "Temporal resolution analysis in frequency domain linear prediction," *The Journal of the Acoustical Society of America (JASA)*, vol. 132, no. 5, pp. EL436–EL442, 2012.

[119] S. Ganapathy, S. Thomas, and H. Hermansky, "Temporal envelope compensation for robust phoneme recognition using modulation spectrum," *The Journal of the Acoustical Society of America (JASA)*, vol. 128, no. 6, pp. 3769–3780, 2010.

[120] B. Wickramasinghe, S. Irtza, E. Ambikairajah, and J. Epps, "Frequency domain linear prediction features for replay spoofing attack detection." in *Proc. Interspeech*, 2018, pp. 661–665.

[121] S. Fernando, V. Sethu, and E. Ambikairajah, "Sub-band envelope features using frequency domain linear prediction for short duration language identification," in *Proc. Interspeech*, 2018, pp. 1818–1822.

[122] S. P. Dubagunta and M. Magimai-Doss, "Using speech production knowledge for raw waveform modelling based Styrian dialect identification," in *Proc. Interspeech*, 09 2019, pp. 2383–2387.

[123] A. Hanani and R. Naser, "Spoken Arabic dialect recognition using x-vectors," *Natural Language Engineering*, vol. 26, no. 6, p. 691–700, 2020.

[124] A. Das, K. Kumar, and J. Wu, "Multi-dialect speech recognition in English using attention on ensemble of experts," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2021, pp. 6244–6248.

[125] S. Shon, A. Ali, and J. Glass, "Convolutional neural network and language embeddings for end-to-end dialect recognition," in *Proc. ODYSSEY*, 2018, pp. 98–104.

[126] Q. Gao, H. Wu, Y. Sun, and Y. Duan, "An end-to-end speech accent recognition method based on hybrid CTC/attention transformer ASR," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2021, pp. 7253–7257.

[127] M. Slaney, "Auditory toolbox," *Interval Research Corporation, Tech. Rep*, vol. 10, p. 1194, 1998.

[128] Q. Yan, S. Vaseghi, D. Rentzos, C.-H. Ho, and E. Turajlic, "Analysis of acoustic correlates of british, australian and american accents," 01 2003, pp. 345 – 350.

[129] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[130] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. on Learning Representations*, 2015.

[131] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[132] Z. Qi, Y. Ma, M. Gu, Y. Jin, S. Li, Q. Zhang, and Y. Shen, "End-to-end Chinese dialect identification using deep feature model of recurrent neural network," in *Proc. Conference on Computer and Communications (ICCC)*, 2018, pp. 2148–2152.

[133] W. Cai, D. Cai, S. Huang, and M. Li, "Utterance-level end-to-end language identification using attention-based CNN-BLSTM," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2019, pp. 5991–5995.

[134] M. Najafian, S. Khurana, S. Shan, A. Ali, and J. Glass, "Exploiting convolutional neural networks for phonotactic based dialect identification," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2018, pp. 5174–5178.

[135] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[136] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv:1803.01271*, 2018.

[137] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[138] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[139] O. Abdel-Hamid, A. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2012, pp. 4277–4280.

[140] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," in *Proc. Association for Computational Linguistics (ACL)*, 2017, pp. 562–570.

[141] S. C. B. Lo, H. P. Chan, J. S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural networks*, vol. 8, no. 7-8, pp. 1201–1214, 1995.

[142] Y. Cui, M. Jia, T. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9260–9269.

[143] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.

[144] V. Peddinti, G. Chen, V. Manohar, T. Ko, D. Povey, and S. Khudanpur, "JHU ASpIRE system: Robust LVCSR with TDNNs, ivector adaptation and RNN-LMS," in *Proc. Automatic Speech Recognition and Understanding (ASRU) Workshop*, 2015, pp. 539–546.

[145] SoX, "Audio manipulation tool," http://sox.sourceforge.net/sox.html, [Online] Available.

[146] A. Waibel, "Modular construction of time-delay neural networks for speech recognition," *Neural computation*, vol. 1, no. 1, pp. 39–46, 1989.

[147] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2015, pp. 3214–3218.

[148] A. Pandey and D. Wang, "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2019, pp. 6875–6879.

[149] S. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. H. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2019.

[150] D. Amodei et al., "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. International Conference on Machine Learning (ICML)*, 2016, p. 173–182.

[151] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. International Conference on Machine Learning (ICML)*, 2006, pp. 369–376.

[152] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[153] Mozilla, "Project Common Voice,[Online]," Available: https://voice.mozilla.org/en/data, 2017.

[154] S. Hema and M. A. Redford, "The effects of native language on Indian English sounds and timing patterns," *Journal of Phonetics*, vol. 41, pp. 393–406, 2014.

[155] S. Ghorbani and J. H. Hansen, "Leveraging Native Language Information for Improved Accented Speech Recognition," in *Proc. Interspeech*, 2018, pp. 2449–2453.

[156] O. Maxwell and J. Fletcher, "The acoustic characteristics of diphthongs in Indian English," *World Englishes*, vol. 29, no. 1, pp. 27–44, 2010.

[157] C. R. Wiltshire and J. D. Harnsberger, "The influence of Gujarati and Tamil L1s on Indian English: A preliminary study," *World Englishes*, vol. 25, no. 1, pp. 91–104, 2006.

[158] A. Prasad and P. Jyothi, "How accents confound: Probing for accent information in end-to-end speech recognition systems," in *Proc. Association for Computational Linguistics (ACL)*, 2020, pp. 3739–3753.

[159] S. B. Kalluri, D. Vijayasenan, S. Ganapathy, R. R. M, and P. Krishnan, "NISP: A multi-lingual multi-accent dataset for speaker profiling," https://github.com/iiscleap/NISP-Dataset, [Online] Available.

[160] Kethireddy, Rashmi, S. R. Kadiri, and S. V. Gangashetty, "Exploration of temporal dynamics of frequency domain linear prediction cepstral coefficients for dialect classification," *Applied Acoustics*, vol. 188, p. 108553, 2020.

[161] ——, "Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations," *The Journal of the Acoustical Society of America (JASA)*, vol. 151, no. 2, pp. 1077–1092, 2022.

[162] M. A. Humayun, H. Yassin, and P. E. Abas, "Native language identification for Indian-speakers by an ensemble of phoneme-specific, and text-independent convolutions," *Speech Communication*, 2022.

[163] Y. Kim, Y. Jernite, D. A. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. the Thirtieth AAAI Conference on Artificial Intelligence*, Nov 2016, pp. 2741–2749.

[164] S. Takase, J. Suzuki, and M. Nagata, "Character n-gram embeddings to improve RNN language models," in *Proc. The Thirty-Third AAAI Conference on Artificial Intelligence*, Sep 2019, pp. 5074–5082.

[165] D. Gerz, I. Vulic, E. M. Ponti, J. Naradowsky, R. Reichart, and A. Korhonen, "Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction," *Trans. of the Association for Computational Linguistics*, vol. 6, pp. 451–465, 2018.

[166] S. Yu, N. Kulkarni, H. Lee, and J. Kim, "Syllable-level neural language model for agglutinative language," in *Proc. the First Workshop on Subword and Character Level Models in NLP*, Sep 2017, pp. 92–96.

[167] Z. Assylbekov, R. Takhanov, B. Myrzakhmetov, and J. Washington, "Syllable-aware neural language models: A failure to beat character-aware ones," *CoRR*, vol. abs/1707.06480, Aug 2017.

[168] M. Tummalapalli and R. Mamidi, "Syllables for sentence classification in morphologically rich languages," in *Proc. the 32nd Pacific Asia Conference on Language, Information and Computation*, 1–3 Dec. 2018.

List of Publications Submitted on the Basis of Thesis

Papers in Journals

1. **Rashmi Kethireddy**, Sudarsana Reddy Kadiri, and Suryakanth V Gangashetty. "Deep neural architectures for dialect classification with single frequency filtering and zero-time windowing feature representations.", The Journal of the Acoustical Society of America (JASA), Vol.151, Issue 2, 2022.

2. **Rashmi Kethireddy**, Sudarsana Reddy Kadiri, and Suryakanth V Gangashetty, "Exploration of temporal dynamics of frequency domain linear prediction cepstral coefficients for dialect classification", Applied Acoustics, pp. 108553, Vol. 188, 2022.

3. **Rashmi Kethireddy**, Sudarsana Reddy Kadiri, Paavo Alku, and Suryakanth V Gangashetty, "Mel-weighted single frequency filtering spectrogram for dialect identification", IEEE Access, pp. 174871–174879, Vol. 8, 2020.

Papers in Conferences

1. **Rashmi Kethireddy**, Sudarsana Reddy Kadiri, Santosh Kesiraju, and Suryakanth V Gangashetty, "Zero-time windowing cepstral coefficients for dialect classification", In Proc. ODYSSEY, pp. 32–38, 2020.

Works to be submitted

1. Leveraging L1 Embeddings for Non-native Indian English ASR, Submission in process to The Journal of the Acoustical Society of America (JASA).

List of other publications

1. **Rashmi Kethireddy**, Sudarsana Reddy Kadiri, and Suryakanth V Gangashetty, "Learning Filterbanks from Raw Waveform for Accent Classification", In Proc. IJCNN, pp. 32–38, 2020.

2. Sudarsana Reddy Kadiri, **Rashmi Kethireddy**, and Paavo Alku, “Parkinson’s Disease Detection from Speech Using Single Frequency Filtering Cepstral Coefficients.”, In Proc. Interspeech, pp. 4971-4975, 2020.

3. VLV Nadimpalli, S Kesiraju, R Banka, **Rashmi Kethireddy**, Suryakanth V Gangashetty, “Resources and benchmarks for keyword search in spoken audio from low-resource Indian languages.“, IEEE Access, pp. 34789-34799, Vol. 10, 2022.

Courses Credited

S.No.	Course Name	Year (Sem)	Grade
1	Statistical Methods in AI	2017 (Monsoon)	A-
2	Database Systems	2017 (Monsoon)	A
3	Speech Technology	2017 (Monsoon)	A
4	Cognitive Neurosciences	2018 (Spring)	A-
5	Optimization Methods	2018 (Spring)	A-
6	Speech Systems	2018 (Spring)	A

Overall CGPA: 9.5/10