HOKKAIDO UNIVERSITY

Title	A Study of Noise-Robust Speech Recognition and Time-Varying Speech Features
Author(s)	Ang, Federico Mendoza
Degree Grantor	北海道大学
Degree Name	博士(情報科学)
Dissertation Number	甲第12190号
Issue Date	2016-03-24
DOI	https://doi.org/10.14943/doctoral.k12190
Doc URL	https://hdl.handle.net/2115/61746
Type	doctoral thesis
File Information	Federico_Mendoza_Ang.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

HOKKAIDO UNIVERSITY

DOCTORAL THESIS

A Study of Noise-Robust Speech Recognition and Time-Varying Speech Features

Author:
Federico ANG

Supervisor:
Dr. Yoshikazu MIYANAGA

Examiners:
Dr. Kunimasa SAITOH
Dr. Takeo OHGANE
Dr. Hiroshi TSUTSUI

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Information and Communication Networks Laboratory
Graduate School of Information Science and Technology

November 2015

HOKKAIDO UNIVERSITY

Abstract

Media Networks Division
Graduate School of Information Science and Technology

Doctor of Philosophy

A Study of Noise-Robust Speech Recognition and Time-Varying Speech Features

by Federico Ang

Noise or variabilities in speech signals take different predominant forms across different languages. Two case scenarios are presented, one where the variability can be solved using data-driven methods and one where an analysis-based hypothesis is necessary. In a data-driven solution, the compensation for mismatch is solved by supplying more data that is representative of the missing information. Such is the case for languages that make use of code-switching, or words from another language as a form of variability.

For noise problems requiring an analysis-based solution, the first line of defense is with the speech features, as it represents the speech input. Because of strict requirements in real-time processing for practical applications, the development of speech features for automatic speech recognition systems were driven by short-time analysis methods. A known limitation of short-time analysis is it fails to capture fast-changing phenomena within a frame of speech as it computes a single output representation. The limitation is further supported by the augmentation of derivative-based features to acquire gains in performance.

As a solution, this work provides a feature extraction framework in which time-varying features are provided that is equal to the number of samples rather than a single representation per frame. Experiments were conducted on a highly acoustic model dependent speech recognition task to reveal issues from analysis of results. It is concluded that in its basic formulation, gains can be acquired by limiting the time-varying extraction only to frames that require accurate modeling, such as signal onsets. This finding results to hybrid systems combining time-varying and time invariant features that can improve the baseline recognition rate for up to an average of 2% including noisy environments.

Acknowledgements

First and foremost, all praises belong to God who has allowed me to exist and experience His mercy and loving kindness and for touching the following people who have helped me along the way:

I extend my deep gratitude to my adviser, Prof. Yoshikazu Miyanaga, who believed in me and supported me as a Ph.D student in all aspects. There were so many things that I did not fulfill as his advisee and I wish to thank him for not giving up on my shortcomings. I also thank Prof. Hiroshi Tsutsui, who has contributed for my professional growth by sharing his insightful comments and ideas for my research and my academic career. I also wish to thank Prof. Kunimasa Saitoh and Prof. Takeo Ohgane for taking time to examine the contents of this thesis.

I thank my fellow researchers from the Philippines who have also contributed to my research on large vocabulary speech recognition namely, Prof. Rowena Cristina Guevara, Prof. Rhandley Cajote, Prof. Joel Ilao, Mr. Michael Gringo Angelo Bayona, and Ms. Ann Franchesca Laguna. I also wish to thank Prof. Alexander Waibel, who allowed my use of the CMU-KIT Speech Recognition Toolkit, JANUS.

I thank the institutions that provided financial aid for my subsistence through scholarship: our graduate school, the KDDI Foundation, and the Japanese government through the Ministry of Education, Culture, Sports, Science and Technology. My brothers in church, who have always extended their support in many forms. Mr. Takashi Manase, for believing in me and trusting me by supporting my living expenses. Mr. Xihao Sun, Mr. Thomas Jeffrey Herber, Mr. George Mufungulwa, and Mrs. Yuki Maeda-Higashi who similarly lent their hands through some financial support.

I thank our general secretary, Ms. Kyoko Ikeda for her invaluable support through helpful reminders and correspondence for meeting submission deadlines. I thank all the members of the Information and Communication Networks Laboratory for the kindness and for providing a wonderful research environment. I thank all of my friends who have shown their moral support through their messages and occasional visits to Sapporo: Ms. Romarie Lorenzo, Ms. Grace Santos, Ms. Allia Donna Go, Mr. Reginald Almonte, Mr. Koichi Kondo, and Mr. JP Rodriguez.

Finally, I wish to thank my family for all the love and support, and for never questioning the path I took.

Contents

Abstract ii

Acknowledgements iii

Contents v

List of Figures viii

List of Tables ix

Abbreviations x

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Background and Objectives . . . 3
- 1.3 Thesis Organization . . . 4

2 A Data-Driven Case Scenario 6
- 2.1 Filipino LVCSR and the Code-Switching Problem . . . 6
- 2.2 System Parameters . . . 7
  - 2.2.1 Speech Database . . . 7
  - 2.2.2 System Front-End . . . 8
  - 2.2.3 Acoustic Modeling . . . 9
  - 2.2.4 Language Modeling . . . 9
  - 2.2.5 Decoding Set-up and Speed . . . 11
- 2.3 Baseline Experiments and Results . . . 12
- 2.4 Experiments on Code-Switching Effects . . . 15
  - 2.4.1 General Observations . . . 15
  - 2.4.2 Error Trends . . . 16
- 2.5 Summary . . . 19

3 The HU-SCS Speech Recognition System 21
- 3.1 Chapter Overview . . . 21
- 3.2 System Dimensions . . . 22
- 3.3 The Speech Recognition System . . . 24

3.4 Front-End Processing . . . 25
3.4.1 Pre-emphasis . . . 26
3.4.2 Windowing and Power Spectrum Computation . . . 27
3.4.3 Mel-Frequency Spectrum . . . 28
3.4.4 DCT on the Log Spectrum . . . 29
3.4.5 Delta Cepstrum . . . 30
3.5 Acoustic Modeling . . . 31
3.5.1 HMM Training . . . 32
3.6 Decoding . . . 36
3.7 Noise Compensation Techniques . . . 37
3.7.1 Cepstral Mean Subtraction . . . 37
3.7.2 Filtering in the Modulation Spectrum . . . 38
3.7.3 Dynamic Range Adjustment . . . 38
3.8 Problem in Baseline System . . . 38

4 Feature Extraction Modifications 40
4.1 Chapter Overview . . . 40
4.2 Short-Time Speech Feature Representation . . . 41
4.3 Time-Varying Linear Prediction . . . 44
4.3.1 General Concepts and Scope . . . 44
4.3.2 Estimating TV-LPC Coefficients . . . 45
4.3.3 Solving TV-LPC Coefficients . . . 47
4.3.4 A Covariance Method Algorithm . . . 49
4.4 Time-Varying Cepstral Coefficients . . . 51
4.4.1 Stability Issues . . . 52
4.5 TV-LPCC Feature Extraction . . . 54
4.5.1 A Running Example . . . 54
4.5.2 Feature Reduction-based Models . . . 56

5 Experimental Setup 60
5.1 Chapter Overview . . . 60
5.2 Experimental Measures . . . 60
5.2.1 Model Selection . . . 60
5.2.2 Training and Testing Scheme . . . 61
5.2.3 Evaluation Metrics . . . 61
5.3 General Setup . . . 62
5.3.1 Database . . . 62
5.3.2 Parameter Settings . . . 62
5.4 Noise Robustness Experiments . . . 63
5.4.1 Noisy Model Generation . . . 63
5.4.2 Noise Compensation . . . 64

6 Results and Analysis 65
6.1 Results and Discussion . . . 65
6.1.1 Recognition Results . . . 65

6.1.2 Discussion of General Findings 68
6.2 Hybrid Models 69
6.2.1 Voting-Based Models 69
6.2.2 Data Selective Models 71

7 Conclusion and Future Direction **76**
7.1 Summary of Thesis 76
7.1.1 Contributions 76
7.1.2 Results Summary 77
7.2 Recommendations for Future Work 78

A Vocabulary List **79**

Bibliography **82**

List of Figures

1.1 Closely sounding speech signals . . . 3

2.1 Self- and cross-coverage plots . . . 10
2.2 Lattice rescoring example . . . 12
2.3 Evaluation report example . . . 13
2.4 Word error rates for the development and evaluation sets of the first written labels . . . 14
2.5 Alignment during evaluation . . . 20
2.6 General error trends . . . 20

3.1 Isolated-word speech recognition system . . . 24
3.2 Cepstral-based feature extraction . . . 26
3.3 Mel filterbank example . . . 29
3.4 Left-to-right HMM topology . . . 31
3.5 Illustration of Gaussian mixture modeling . . . 33
3.6 Baseline system confusion matrix . . . 39

4.1 AR model for speech synthesis . . . 44
4.2 Pole-zero diagram of an unstable filter . . . 54
4.3 Speech signal for TV-LPCC analysis . . . 55
4.4 FFT and spectral estimate from LPC . . . 55
4.5 Line evolution spectrum from TV-LPCC coefficients . . . 56
4.6 High resolution spectral estimate from TV-LPCC . . . 57
4.7 Feature reduction schemes . . . 58
4.8 TV-LPCC clusters based on SPDIFF as a correlation metric . . . 59

6.1 Effect of VAD performance to cepstrum trajectories . . . 68
6.2 Voting-based scheme . . . 70
6.3 Modified HMM topology . . . 71
6.4 Experiment on closely sounding words using 32-state HMM . . . 71
6.5 Experiment on closely sounding words using 16-state skipping HMM 72
6.6 Voting-based system . . . 73
6.7 Near sounding word recognition as a function of number of frames used . . . 74
6.8 Split signal data-selective model scheme . . . 74

List of Tables

2.1 Filipino speech data statistics . . . 8
2.2 Feature extraction summary . . . 9
2.3 Language model perplexities on test set . . . 11
2.4 Best WERs of Filipino ASR system from successive label writing . . . 14
2.5 Updated performance of trained systems (WER) . . . 15
2.6 Average relative percentage of word types to insertions and deletions (pre-filtering) . . . 16
2.7 Average absolute percentage contributions to WER (in parentheses) of Tagalog versus loan words (development set only) . . . 16
2.8 Substitution error types and trends in order of relative frequencies of occurence (pre-filtering) . . . 17
2.9 Ratio of utterances with and without specific language conditions . . . 18
2.10 Ratio of number of reference words per set . . . 18
2.11 WERs of specific language condition sets in the pre- and post-filtering stages . . . 19

5.1 Speakers used for cross-validation . . . 61
5.2 System summary . . . 63
5.3 NOISEX noise types . . . 63
5.4 Reduction in accuracy rates for frequency-weighted mixing of noise (negative values indicate improvements) . . . 64

6.1 Average results for clean experiments . . . 65
6.2 MFCC baseline (male) . . . 66
6.3 MFCC baseline (female) . . . 66
6.4 TV-LPCC skipping (male) . . . 66
6.5 TV-LPCC skipping (female) . . . 67
6.6 TV-LPCC averaging (male) . . . 67
6.7 TV-LPCC averaging (female) . . . 67
6.8 Average results for voting-based experiment . . . 72
6.9 Average results for hybrid experiments . . . 75

Abbreviations

AR	**A**uto**R**egressive
ARMA	**A**uto**R**egressive-**M**oving **A**verage
ASR	**A**utomatic **S**peech **R**ecognition
CMN	**C**epstral **M**ean **N**ormalization
CMS	**C**epstral **M**ean **S**ubtraction
CV	**C**onsonant-**V**owel
CVN	**C**epstral **V**ariance **N**ormalization
DCT	**D**iscrete **C**osine **T**ransform
DMP	**D**iscrete **M**arkov **P**rocess
DRA	**D**ynamic **R**ange **A**djustment
FIR	**F**inite **I**mpulse **R**esponse
FSA	**F**eature **S**pace **A**daptation
FT	**F**ourier **T**ransform
GMM	**G**aussian **M**ixture **M**odel
HMM	**H**idden **M**arkov **M**odel
HU-SCS	**H**okkaido **U**niversity **S**peech **C**ommunication **S**ystem
IDFT	**Inverse D**iscrete **F**ourier **T**ransform

IFT	**I**nverse **F**ourier **T**ransform
LM	**L**anguage **M**odel
LDA	**L**inear **D**iscriminant **A**nalysis
LPC	**L**inear **P**redictive **C**oding
LPCC	**L**inear **P**redictive **C**epstral **C**oefficients
LVCSR	**L**arge **V**ocabulary **C**ontinuous **S**peech **R**ecognition
MFCC	**M**el-**F**requency **C**epstral **C**oefficients
OFS	**O**ptimal **F**eature **S**pace
OOV	**O**ut-**O**f-**V**ocabulary
PARCOR	**PAR**tial **COR**relation
PLP	**P**erceptual **L**inear **P**rediction
RASTA	**R**el**A**tive **S**pec**T**r**A**
ROVER	**R**ecognizing **O**utput **V**oting **E**rror **R**eduction
RSA	**R**unning **S**pectrum **A**nalysis
RSF	**R**unning **S**pectrum **F**iltering
SNR	**S**ignal-to-**N**oise **R**atio
SS	**S**pectral **S**ubtraction
STC	**S**emi-**T**ied **C**ovariance
TV	**T**ime-**V**arying
VAD	**V**oice **A**ctivity **D**etection
VC	**V**owel-**C**onsonant
VTLN	**V**ocal **T**ract **L**ength **N**ormalization
WER	**W**ord **E**rror **R**ate

Chapter 1

Introduction

Speech communication is a very important aspect of human interaction, as it is the way for people to establish social bonds, express ideas, and exchange information. With the advent of technological advancement, it is only but natural for humans to attempt developing automated systems and applications that would enable computers to mimic, or even do something more efficient than, what humans can do when they perceive speech. This is the goal of the field of automatic speech recognition or abbreviated as ASR.

1.1 Motivation

Speech recognition system development during its earlier days were guided by research that is coupled by multidisciplinary collaborations, making use of human perception as a guide for modeling speech processes. As time passed, as developments were being made to speech recognition systems with varying degrees of complexity, certain limiting aspects of development became widespread as dictated by the technologies that were proven effective in driving the improvements in performance. For example, current speech recognition technology relies on data-driven methods of machine learning and many are convinced that speech database quantity and quality have big roles to play in speech recognition system development. While data-driven approaches are gaining popularity due to improvements in processing technology, which enabled complex artificial neural networks and

deep learning methods, theory-driven analysis can help in the aspects of refining the limits and boundaries of complex systems.

The same situation can be said about the use of well-established speech feature extraction methods. The Mel-frequency cepstral coefficient or MFCC, for example, became a staple in most commercial ASR systems and considered as a practical choice for most development. The work in this thesis is motivated by certain shortcomings of the MFCC, specifically when used in a limited architecture, using whole-word hidden Markov model (HMM) units that is suitable for small-scale ASR hardware design. This defect of the MFCC is due to the quasi-stationary assumptions imposed by short-time processing that is inherent in most perceptually motivated feature extraction methods. This is also supported by the fact that the use of MFCC from word-level to continuous speech recognition is always accompanied by the computation of differentials, reflecting dynamic information in transitions. To be able to capture CV or VC transitions in the spectral domain is an important aspect of speech modeling since it often defines the difference of a word from other words. The differences, however, cannot be easily detected from the temporal signal.

Take for example, the two speech signals shown in Figure 1.1. The uttered words in these two signals are "genki" and "denki" in Japanese. If the signals shown in the bottom graphs of the figure are to be processed by a nonstationary method, the fast changing phenomena that defines the difference between the two words will be lost. This stationary assumption is relaxed when the underlying speech model of the features is allowed to be nonstationary, effectively capturing transient parts of the speech signal that often differentiates the words being recognized.

Aside from the effectiveness of the time-varying speech feature as a speech model representation, the inherent robustness to speech variability is also a main concern [1]. In the literature, modeling the nonstationarity of speech signals is already considered as a way to address variability [2, 3]. Other research delved on the use of feature compensation techniques where mismatches in training and testing data are analyzed and resolved. Some of these techniques are already considered as staple in the speech recognition field and their interaction with the proposed speech features will also be investigated in this thesis. Another motivation for this research is that the use of time-varying speech features also falls in between the

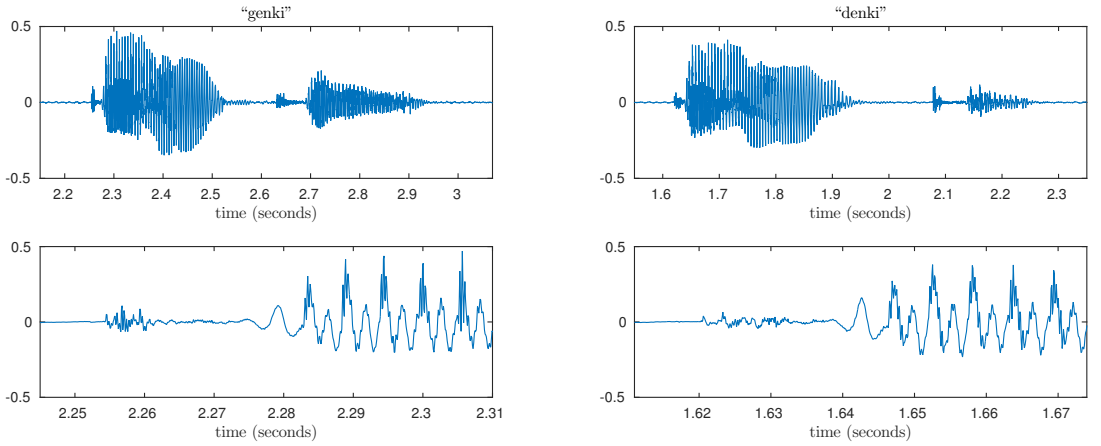

FIGURE 1.1: Closely sounding speech signals "genki" and "denki" and onsets.

concept of additional cues and features for enhancement coupled with dimensionality reduction and feature selection. This is because time-varying techniques are considerably high in resolution compared to the time-invariant counterparts.

1.2 Research Background and Objectives

The research work done in this thesis is based on an on-going collective research effort aptly called the Hokkaido University speech communication system (HU-SCS) [4], which aims to continuously develop and improve speech communication technologies. This system integrates speech detection, robust speech analysis, speech recognition, and speech rejection to create practical systems that can be implemented on hardware. In its current stage of development, it has already released consumer electronics that utilize these speech technologies under the guidance of fast and low power hardware design [5]. Through this, the position of this thesis in the entire speech recognition field can be clarified through the limitations imposed by the target speech recognition system. In particular, the enhancements that the work in this thesis aims to show is only signal-based and will not affect the computationally efficient hardware design that is already in place.

It is the concern in the research done for this thesis that the current state of isolated-word, ASR development be pushed on one side of its field. It is concerned with presenting theoretical and experimental details on the use of time-varying speech features, specifically for enhancing speech recognition system performance

under noisy conditions. By noise, it only does not mean noise associated with the speech production mechanism of humans, but also external factors and overall variability in speech production, acquisition, processing, and storage [2]. Although these variabilities are not targeted individually, the aim for development is to test for the inherent robustness of the speech feature when the overall variability is lumped together at the recognition stage. Because of this aim, the target platform will be a speech recognition system whose performance is highly dependent on the accuracy of the speech features, which represent the speech model.

The following are the objectives for undertaking this thesis:

1. To highlight the different approaches to dealing with speech variability for speech recognition by providing test case scenarios.
2. To illustrate the problems inherent in using stationary speech features in a practical setting.
3. To present a practical approach in applying time-varying speech feature extraction methods for speech recognition systems and determine possible ways of using these features.
4. To investigate the performance of time-varying LPC-based features under different noise conditions. This is due to the lack of experimental findings in the subject. Most time-varying speech modeling applications are concerned with spectral estimation and in solving the underlying problems of the model itself, and experimental findings for speech recognition are few.
5. Present a working isolated-word speech recognition system that makes use of time-varying speech features and prove its model to be effective.

1.3 Thesis Organization

Chapter 2 first provides a different test-case scenario that highlights the common approach to dealing with errors in speech recognition. This chapter showcases the development of a large vocabulary continuous speech recognition (LVCSR) system where typically, data-driven methods are used. The discussion in this chapter will

serve as a contrast to an analysis-based method that will become the main focus of the thesis.

Chapter 3 gives an overview of isolated-word speech recognition systems. It hinges on the common notion that speech recognition systems are highly application dependent. The aim of this chapter is to present the baseline architecture of the isolated-word speech recognition system that serves as the backbone for the analysis-based focus of this thesis. While discussing the building blocks of the speech recognition system, it is also an opportunity to differentiate the techniques used in other systems in the literature. Thus, this chapter will also be a review of the fundamentals in ASR while highlighting the limits of the baseline system. However, procedures that only apply to continuous speech are only mentioned slightly and will not be discussed in detail, unless the adoption of techniques introduced in this thesis will immensely affect the said procedures for continuous speech. Aside from discussing the details of the working architecture, this chapter also summarizes the evaluation efforts and the currently known performance of the said baseline system. This allows for a shift of the discussion to the existing problems and the proposed solution, which will be the subject of the succeeding chapter.

Chapter 4 details the ideas and adoption of several methods that leads to the use of the proposed time-varying speech features. The various implementations based on different criteria will also be discussed, as well as possible issues, used solutions, and justifications that were made.

Chapter 5 deals with the practical aspects of setting up the experimental environment for conducting the evaluation of the different models. This includes system configuration parameters, details of the database and its division, and creation of noisy models for robustness tests. Confidence measures and evaluation metrics, as well as bias-variance considerations are discussed in this chapter.

Chapter 6 presents the results of the experiments and model selection procedures made. Post-experiment analysis and additional work are also discussed in this chapter.

Chapter 7 concludes the thesis with a summary of results, contributions, and recommendations for future work.

Chapter 2

A Data-Driven Case Scenario

2.1 Filipino LVCSR and the Code-Switching Problem

LVCSR research for Filipino, the official language of the Philippines, can be considered as relatively few in the speech recognition literature [6]. One possible reason is the lack of interest due to its ongoing state of standardization and intellectualization that in order to develop a system, researchers will be forced to rely on statistical information and base the description of Filipino on what is in widespread use. While a Filipino speech database has been collected in the past for the purpose of speech recognition[7], a large percentage of its contents was found to be pure Tagalog, the former national language. This is due to the fact that Tagalog and Filipino share identical grammar based on linguistic rules[8]. This led to a degradation in the recognition performance when actual Filipino sentences are spoken. Practical Filipino sentences as input degrades the performance of older systems by 40-50%. Thus, there is a need for a development of a Filipino ASR system that reflects the true character of the language. For one, the Philippines is well-known as an English-speaking nation. Most Filipinos speak English as an additional language for daily, casual conversations.

In linguistics, using two or more languages in the context of a single conversation is commonly known as *code-switching*. In the context of speech recognition, this

means that all frequently used English words must be added in the lexicon and that the language model must accommodate the probabilities for these words as well. Thus, addressing code-switching in Filipino ASR is to deal with the combined problems and inherent suboptimalities of the two languages involved. As there exist no established evaluation results for this type of speech recognition task, this research was initiated, including the parameters and settings that come with data-driven development. As with other published research on Filipino ASR[9–11], the work outlined in this chapter is by no means exhaustive. Details of the text data for language modeling were also evaluated, avoiding the data sparsity and bias issues previously reported using an older Filipino database[12].

2.2 System Parameters

For the evaluations to follow, an explicit analysis of the ASR channel conditions was not considered. Thus, there is a need to clarify the parameters and the dimensions of the system.

2.2.1 Speech Database

About 15000 prompts that contained Filipino isolated words, phrases, and sentences were used as reading materials. Spontaneous speech were also recorded where the speakers were asked about several random topics. The transcriptions for these were manually made with the help of native speakers. All other prompts are automatically associated as the transcripts of the generated recording file. The prompts were taken from a variety of sources with different domains, ranging from news, literary works, daily and situational conversations. These were either downloaded from the internet or taken with permission from digital publications. Spontaneous speech was also recorded at the end of each session. No consideration for phonetic balancing was made in order to reflect the inherent nature of the texts. Each utterance recorded is limited to a span of one minute.

All speech data were recorded at a high rate and down-sampled to 16-kHz, at 16-bit resolution per sample in a single channel. Two set-ups were made, depending on whether the activity was done in the lab or off-site. Recordings inside the laboratory were done inside a pseudo-anechoic chamber while the off-site recording required a much simpler set-up, involving only a headset directly connected to a laptop. About 30% of the recordings were made under moderately noisy environments (between 15 and 20 dB SNR) for noise compensation.

Because the speech corpus and the recognition system were simultaneously being developed at the start of the research, only a subset of the speech corpus described was used. This subset contains An additional set of 25 male and 25 female speakers were added later due to a large number of overlapping prompts in the original set. To compensate for moderately noisy environments, additional speech data containing such channel conditions were used. These additional data consisted of 25 male and 25 female speakers. These data were also collected due to a large number of overlapping prompts in the original set. A summary of the resulting division for the development, test, and evaluation sets is given in Table 2.1.

TABLE 2.1: Filipino speech data statistics

	Training	**Testing**	**Evaluation**	**Total**
Speakers (unique)	146 (144)	5	5	156 (154)
Utterances	33,340	1,525	1,527	36,392
Prompts covered	12,474	1,498	1,509	15,481
Words	375,039	19,961	19,100	414,100
Vocabulary	14,461	4,384	4,163	15,673
Duration in hours	54.9	2.8	2.8	60.6

2.2.2 System Front-End

The set-up for feature extraction is summarized in Table 2.2. The computation of derivatives for temporal dependency before linear discriminant analysis (LDA) is done via a linear transformation using 15 frames around the current frame. Vocal tract length normalization (VTLN)[13] is also applied on a per speaker basis as an enhancement.

TABLE 2.2: Feature extraction summary

Features	Mel-Frequency Cepstral Coefficients
Frame length	16 msec
Overlap length	6 msec
Mel-filters	30
Normalization	Mean, variance
Pre-LDA dimension	240
Post-LDA dimension	48
VTLN	linear domain

2.2.3 Acoustic Modeling

Our system is HMM-based (three-state Bakis model) and only fully continuous systems are considered. The distributions were trained using recursive Gaussian splitting [14] with a maximum of 64 Gaussians, and improved by a variant of semi-tied covariance (STC) [15] training called optimal feature space (OFS) training. This latter training results in a global invariant transformation matrix [16] that incorporates the LDA matrix computed previously. We bootstrapped a small English seed model and used it to generate initial labels for the training data. A context independent system was trained and used to bootstrap a quintphone-based context dependent system. There were 54 manually-created classes for the phone set for state tying.

2.2.4 Language Modeling

Before actual language modeling was done, some analysis were made via scripting. Aside from the sparsity of the training text due to the overlapping utterances across speakers, it was also observed that training text for language modeling cannot be included because of its high correlation to the testing set utterances. We prepared subsets of texts from the remaining prompts not covered by the database then correlated the texts to the test corpus via a maximum likelihood-based weighted interpolation scheme. For all cases, the training corpus constitutes the mixture by 99%. Because of this, the training set was instead considered as the heldout set. After removing the overlaps with the testing set, it was used as the tuning set for LM training. Due to the high correlation of the training text to the test set utterances, three different text sets obtained from the internet through

a crawler were prepared for the experiments. Figure 2.1 shows how the self- and cross-coverage of the training set are very close to each other while the texts from the crawler are more variable. Note that for the vocabulary of the corpora from the internet of 360k words, the OOV rate for the test set was still at 3.44%.

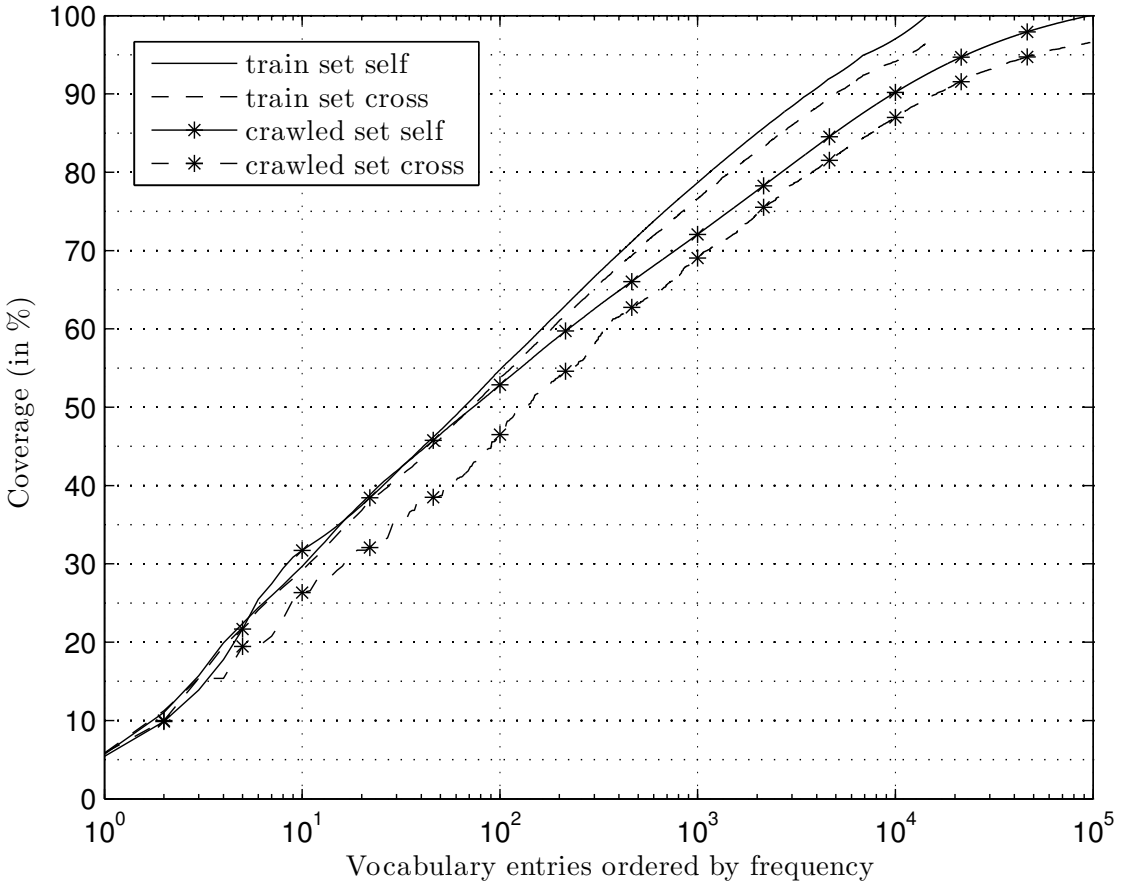

FIGURE 2.1: Self- and cross-coverage plots of the training (train) set and the 13 Million (crawler) words set to the test set.

The vocabulary for the LMs was generated by taking the most frequent words from the corpora using the crawler before a cutoff and crossing it with the words from the training set. This provided a 10.5k vocabulary, with an OOV of 5.45% when compared to the vocabulary from the test corpus.

Using the three subcorpora from the web, 4-gram back-off and interpolated models for several smoothing techniques [17–22] were generated. For each smoothing algorithm, the three LMs $P_1(w|h)$, $P_2(w|h)$, and $P_3(w|h)$ were combined linearly

to generate an interpolated model $P(w|h)$. This was done linearly via

$$P(w|h) = \lambda_1 P_1(w|h) + \lambda_2 P_2(w|h) + \lambda_3 P_3(w|h) \tag{2.1}$$

where interpolation weights λ_1, λ_2, and λ_3 were chosen to maximize the likelihood of the tuning set. After generating all models, the one that minimizes the perplexity over the test set was picked. Table 2.3 shows the results of the training. While differences are subtle for most cases, it is clear that the modified Kneser-Ney algorithm still gives the lowest perplexity. However, while most systems get better results from interpolated n-gram models, ours got better perplexities from the simple back-off models.

TABLE 2.3: Language model perplexities on test set.

Smoothing	**Back-off**	**Interpolated**
Natural (Ristadt)	341.24	–
Good-Turing/Katz	272.52	–
Witten-Bell	268.48	369.14
Absolute (Ney)	267.75	375.48
Orig. Kneser-Ney	267.73	294.68
Mod. Kneser-Ney	**266.94**	279.66

2.2.5 Decoding Set-up and Speed

Lattice rescoring (Figure 2.2) based on language model weight z and word transition penalty p variations is done to search for the necessary statistical correction for the combined acoustic and language model (log) scores:

$$P(W|\mathbf{X}) = P(\mathbf{X}|W)P(W)^z p^{|W|} \tag{2.2}$$

where $P(\mathbf{X}|W)$ is the acoustic model probability, $P(W)$ is the language model probability, and $|W|$ is the length of the utterance. The standard Word Error Rate (WER) is used to evaluate all systems. An example evaluation report is shown in Figure 2.3.

Decoding parameters such as beam settings were heuristically determined from previous experiments that balance recognition accuracy and speed. Feature Space Adaptation (FSA) [23] was applied in decoding based on empirical results where

it consistently gives significant improvements to our systems. The gain for this advantage, however, was inversely proportional to system improvement. The speed of our tool's single pass decoder [24] is at a real-time factor average of 0.13 and 0.21 using the best decoding parameters, for context-independent and context-dependent systems respectively, using a 3.6-GHz Intel Core i7-3820 computer. This fast decoding enables us to use the systems for real-time applications.

2.3 Baseline Experiments and Results

The bootstrapped model was evaluated on the test set and initial results were at 51.5% and 58.6% WER, for the FSA-based and non-FSA decoding respectively. From successive evaluations it was observed that using FSA gives around 3% average advantage over its non-FSA counterpart. A set of recognition scores from the first labeling epoch are given in Figure 2.4 for both development and evaluation sets. From this set of results, seven Viterbi iterations were decided for succeeding evaluations. In the next set of tables, only FSA-based results are shown. Table 2.4 summarizes the results of the successive label writing procedure done using the development training and test sets under a comparable set up of seven Viterbi

```
============================================================
SUMMARY - WER (del,ins)
============================================================
LV      : 30.6 (2.3, 7.3)
FINIS   : 26.4 (2.5, 5.5)
CONS_ALL:  (, )
lz\lp   20              25              30              35
20      28.0(2.6 5.9)   27.9(2.9 5.4)   27.9(3.2 5.2)   27.8(3.5 4.8)
25      26.8(2.6 5.4)   26.8(2.9 5.0)   26.9(3.2 4.8)   27.0(3.5 4.5)
30      26.2(2.7 5.2)   26.2(2.9 4.9)   26.3(3.2 4.6)   26.4(3.4 4.5)
35      26.1(2.8 5.0)   26.1(3.0 4.8)   26.1(3.2 4.6)   26.1(3.4 4.4)
40      26.1(2.9 5.0)   *25.9(3.1 4.7)  26.0(3.2 4.5)   26.1(3.5 4.4)
45      26.0(2.9 4.9)   26.0(3.1 4.8)   26.0(3.3 4.6)   26.0(3.4 4.5)
50      26.1(3.0 4.9)   26.1(3.2 4.8)   26.1(3.3 4.6)   26.1(3.4 4.5)
```

FIGURE 2.2: Lattice rescoring example

```
DETAILED OVERALL REPORT FOR THE SYSTEM: H_35_40_*_*.ctm
SENTENCE RECOGNITION PERFORMANCE
 sentences                                          813
 with errors                          70.4%   ( 572)
   with substitions                   66.8%   ( 543)
   with deletions                     19.4%   ( 158)
   with insertions                    31.5%   ( 256)
WORD RECOGNITION PERFORMANCE
Percent Total Error       =   27.7%   (2035)
Percent Correct           =   77.8%   (5721)
Percent Substitution      =   19.1%   (1407)
Percent Deletions         =    3.1%   ( 227)
Percent Insertions        =    5.5%   ( 401)
Percent Word Accuracy     =   72.3%
Ref. words                =           (7355)
Hyp. words                =           (7529)
Aligned words             =           (7756)
```

FIGURE 2.3: Evaluation report example

iterations.

For the context-dependent system, variations were made on the number of distributions. As can be observed, for both cases the best results were achieved from the second writing of labels and performance degrades after subsequent writings. After further viterbi training, the lowest WER of 30.0% and 21.7% were achieved for the context independent and dependent systems, respectively. It is worth noting that the variations made do not have any large impact on the real time performance of the system.

Table 2.5 shows the summary of the performance of the best trained systems and further enhancements to both the test and evaluation sets. From these results the reliability of the performance to unseen data can be deduced as the evaluation results are fairly consistent. Context-dependence gives a huge average advantage

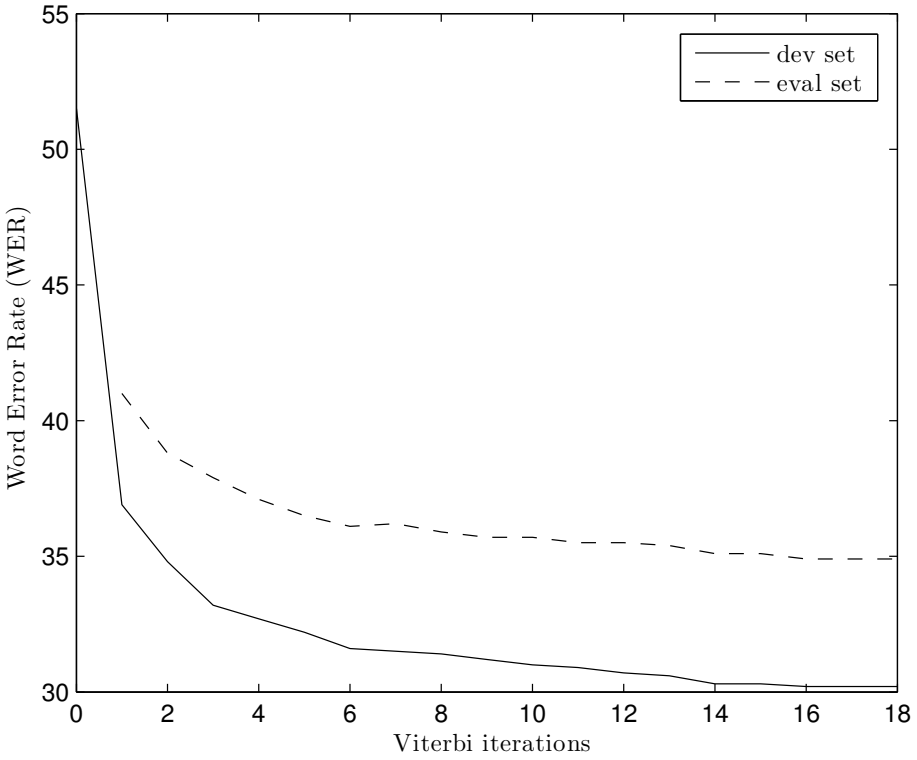

FIGURE 2.4: Word error rates for the development and evaluation sets of the first written labels.

TABLE 2.4: Best WERs of Filipino ASR system from successive label writing.

	Context-independent		Context-dependent	
Labeling epoch	**LDA only**	**OFS**	**2000**	**3000**
1	34.8	31.5	22.5	20.7
2	**33.8**	**30.1**	**22.2**	**20.5**
3	34.0	30.3	22.3	20.7
4	34.3	31.0	22.7	20.9
5	34.4	31.0	23.4	21.2
6	34.6	30.7	23.7	21.4
7	35.0	30.9	24.1	21.9

of around 11%. VTLN, manual corrections of lexical entries and post-filtering of evaluation hypotheses based on frequency errors bring down the WER at a current best performance of 25.3% for the evaluation set. The mappings in the post-filter were decided based on initial substitution error reports. Included in the mapping are the following:

1. Words in Tagalog that are closely pronounced to their English counterparts are mapped to English.

2. Most common spelling variants are mapped to a single consistent spelling.

3. Affixated words with simple root extraction (e.g. *pinaka* + [adjective]) are separated from their affixes. Cases of infixes and more complex inflections are not touched.

4. Contractions are expanded.

TABLE 2.5: Updated performance of trained systems (WER)

Description	**Test**	**Eval**
Context-independent	30.0	35.3
Context-dependent	21.7	27.0
VTLN	21.0	25.9
Manual corrections	**20.5**	**25.3**

2.4 Experiments on Code-Switching Effects

2.4.1 General Observations

Using the system that generated the best performance from Table 2.5, the results of the scoring were analyzed. On average (development and evaluation sets), the distribution of substitutions, insertions, and deletions are roughly at 70%, 15%, and 15%, respectively. A summary of average relative contributions of different word types for insertion and deletion errors can be found in Table 2.6. Note that because there are certain words that can be considered as both a Tagalog or an English word, the total when added up exceeds 100%. Error analysis reports currently do not provide the context of the error occurrence. Based on this summary, loan words contribute up to 26.7% of insertion and deletion errors, which is about 7.8% of the total WER.

The best system with the first FSA-based baseline system were then compared in terms of the absolute contribution of loan words to the WER. From Table 2.7,

TABLE 2.6: Average relative percentage of word types to insertions and deletions (pre-filtering).

	Insertions	**Deletions**
Tagalog	74.6	82.9
Loan words	34.5	17.9
Acronyms	0.6	4.0
Nonspeech	1.1	1.0

the data-driven approach solved roughly 44.8% and 6.6% of the contributions of Tagalog and loan words to WER, respectively. For these systems, this is a 3% absolute increase in accuracy for loan words. This is based on a database that contains 20% of loan words in the lexicon and roughly 40% in the actual prompts.

TABLE 2.7: Average absolute percentage contributions to WER (in parentheses) of Tagalog versus loan words (development set only).

	Baseline (51.5%)	**Best System (22.9%)**
Tagalog	40.1	17.0
Loan words	10.4	7.0

2.4.2 Error Trends

Aside from the fact that the majority of errors are substitution types, the frequency of occurrence and the acoustic relationship between error pairs give more meaningful insights. Based on the evaluations, six major trends of substitution errors were defined. The rank and relative contributions of these are summarized in Table 2.8. From this table, substitution errors that contain loan words on average contribute roughly 7.4% to the total WER. Notable trends in the errors that cannot be simply solved using post-filtering of the hypotheses are mostly morphological in nature. A majority of similar onset errors contain suffixations of *-ng* to nouns that do not contain them (*e.g. ano* becomes *anong*). There is also a prevalence of reduplication of initial and middle syllables for verbs (*e.g. inisip* becomes *iniisip*). Homonyms are mostly concentrated on loan words and can be attributed to poor coverage of the language model. Homonyms can also span more than a single word. Several examples in Tagalog are: *naaakyat* (possible to go up) vs.

na aakyat (that is going up), *naman niyang* vs. *naman 'yang* (these are of different meanings). Orthographical errors due to spelling variants can be alleviated by creating post-filters that can assign a single reference word to multiple entries that have the same connotation. Miscellaneous errors come from acronyms and non-speech sounds.

TABLE 2.8: Substitution error types and trends in order of relative frequencies of occurence (pre-filtering).

Description	**Frequency**	**Example**
Tagalog-Tagalog	65.5	*mula* vs. *wala*
Tagalog-Loan	22.0	*atensyon* vs. *retention*
Loan-Loan	14.0	*adjust* vs. *jazz*
Similar ending	14.4	*wrap* vs. *sarap*
Similar onset	12.9	*akin* vs. *aking*
Homonyms	9.6	*bakit* vs. *bucket*
Different middle	4.5	*buto* vs. *boto*
Orthographical	3.6	*kaunti* vs. *konti*
Others	3.2	*T.V.* vs. *T.B.*

To further investigate the nature of the errors, the decoding of the test set utterances were divided based on some criteria. For *code-switching*, a list of loan words were used to flag each utterance. However, utterances with mere proper noun usage are not considered as a switch. Flagging of utterances with inflections is based on the 80 known affixation rules in Filipino. Particles usage is based on the following words: *na*, *pa*, *man*, *nga*, *din/rin*, *lang*, *naman*, *daw/raw*, *po/ha*, *ba*, *pala*, *muna*, *yata* and some cases of *kaya*, *tuloy*, *kasi*, *sana*. For flagging usage of non-standard orthography, the results from [25] were used for the list of spelling variants. However, the counts were not explicitly included due to the fact that almost all utterances were being included in the set and those being left out from the smaller set are very short phrases. Tables 2.9 and 2.10 give us the result of these counts.

We used the optimal setting for the development set to evaluate all the divided sets. Table 2.11 shows the achieved WERs with and without post-filtering. It is evident that post-filtering the hypotheses gives an average of 0.62% absolute

TABLE 2.9: Ratio of utterances with and without specific language conditions.

Set Condition	**Train**	**Dev**	**Eval**
All utterances	33340	1523	1529
Code-switching	7308 : 26032	552 : 971	560 : 969
Inflections	26886 : 6454	1194 : 329	1205 : 324
Particles usage	17458 : 15882	766 : 757	774 : 755

TABLE 2.10: Ratio of number of reference words per set.

Set Condition	**Dev**	**Eval**
All utterances	20272	19406
Code-switching	11804 : 9669	10131 : 10460
Inflections	18341 : 2730	17635 : 2763
Particles usage	14180 : 7209	13280 : 7355

improvement for all systems. Based on these results, the following observations were made:

1. Generally speaking, around 20% average absolute contribution to the WERs come from uncorrected human errors in the data both from the recordings and the transcriptions, differing accents, and those enumerated in Table 2.8.

2. For code-switching, despite utterances flagged as non-switching being higher in number, it achieved a lower WER due to a lesser number of loan words. Loan words contributed about 6.4% and 0.5% average absolute WER to the switching and non-switching sets, respectively.

3. Utterances with inflections had the highest gain from post-filtering due to the large contribution of Tagalog words. Some inflected words still contributed around 1.82% and 0.73% average absolute WER to the respective cases. These generally come from inflected reference words with reduplicated middle syllables substituted with no reduplication (e.g. *makakaramdam* vs. *makaramdam*). Note however that 9% absolute WER for utterances without inflections come from loan words.

4. The 1.5% average relative advantage of utterances without particles and auxiliary words mostly come from loan words and the general errors. The average absolute contributions of the particles to the WERs are at 0.85% and 0.26%, respectively. Note the difference in number of reference words.

TABLE 2.11: WERs of specific language condition sets in the pre- and post-filtering stages.

Condition	Pre-Filtering		Post-Filtering	
	Dev	Eval	Dev	Eval
All utterances	21.0	25.9	20.5	25.3
Code-switching	24.2 : 17.7	29.7 : 22.8	23.7 : **17.1**	29.0 : **22.2**
Inflection	20.0 : 31.0	24.9 : 35.0	**19.1** : 30.4	**24.1** : 34.4
Particles usage	20.9 : 22.0	25.7 : 27.6	**20.3** : 21.5	**25.0** : 27.1

Finally, the quality of the context-dependent models being generated was investigated through the reports generated by the scoring toolkit. For example, as can be seen in Figure 2.5, statistics of phonetic replacements can be inferred. Due in part to the large number of context-dependent models and the influence of the language model, our analysis was based on the recognition of the training set itself and the language model scores were discounted from the decoding. Not surprisingly, as was observed from the pre-analysis stage of our development, n and ŋ are the most misrecognized. As for vowels, the o and u sounds are the most interchanged. For our systems, solving the distinction problem between n and ŋ can provide around 3.84% absolute increase in recognition accuracy. The vowels u and o accounts for 2% of our systems WER. All other phones are of average contributions to the error. Some however are insignificant in contribution due to underrepresentation. These phones are (in increasing order of representation): z, ɲ, θ, l̪, ð, and d͡ʒ (using IPA symbols). This reflects a subset of the loan words that is accountable for almost 6% of our system's WER. A graph on Figure 2.6 showing specific trends accounting for more than 4% of the average error is also provided as a clearer reference for the limitation of the system.

2.5 Summary

After several training and enhancements, a lowest word error rate of 20.5% was achieved, using a context-dependent system that can still perform in real time. This is at par with the highest reported accuracy for Filipino speech, notwithstanding the fact that our system allows for code switching that contributes considerable difficulty in the recognition. The open-domain language model, speaker-independent acoustic models, and near 80% word recognition accuracy, all point

```
id: (Suppressed)
Labels: <o,f0,female>
File: (Suppressed)
Channel: 1
Scores: (#C #S #D #I) 6 5 1 4
REF:  aalamin KO lamang ANG kaniyang VITAL   signs at ****** ** ...
HYP:  aalamin PA lamang *** kaniyang BAITANG signs at MONTHS AS ...
Eval:         S         D            S                 I      I  ...
REF:          KO                     VITAL             MAGSAGAWA ...
HYP:          PA                     BAITANG           MONTHS AS ...
Eval:         S         D            S                 I      I  ...
REF:        K OX            V AX Y T AX XL   M AX G    S AX      ...
HYP:        P AX            B AX Y T AX NG   M AH N TH S AX S AX...
REF:        K OX            V            XL     AX G              ...
HYP:        P AX            B            NG     AH N TH      S    ...
            S S             S            S      S  S I       I    ...
```

FIGURE 2.5: Alignment during evaluation

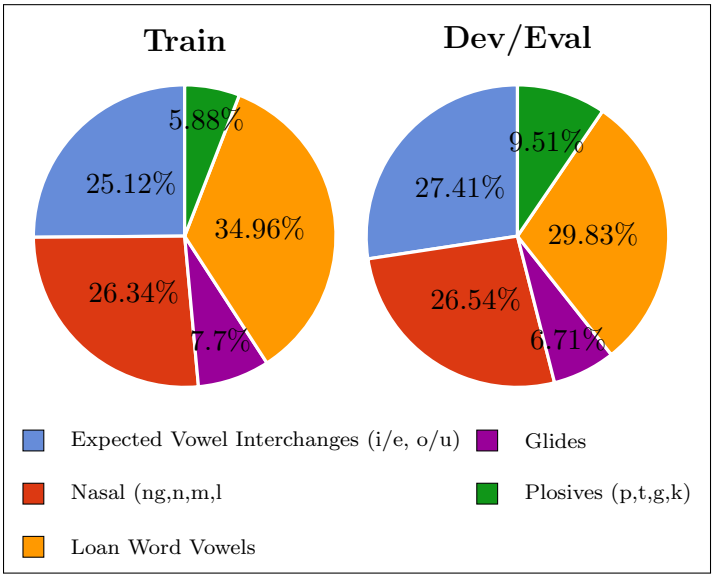

FIGURE 2.6: General error trends

towards the possibility of using it for practical purposes.

Chapter 3

The HU-SCS Speech Recognition System

3.1 Chapter Overview

Automatic speech recognition (ASR) is a technology that allows computers to make sense of human speech as inputs to a machine. This technology helps machines in converting human speech into a machine-readable information that can be further processed to do useful tasks. Nowadays, there are many applications that make use of ASR such as hands-free (voice-operated) controllers, automatic dictation software, and voice-controlled personal assistants in smartphones.

Despite the many useful applications, ASR is a difficult problem because it has to deal with different situations that are often not governed by strict rules. Therefore, before undertaking any study related to ASR, one has to define the target dimensions of the application.

Several approaches to obtain speech features are available in the literature. The three most popular are the Mel-frequency cepstral coefficients, perceptual linear prediction coefficients, and the linear predictive coding cepstral coefficients. The speech features used by the baseline system is the MFCC and its inner workings are explained in this chapter.

To utilize the speech feature representation that is extracted from the speech signal, an acoustic modeling block is necessary. The most popular of which is

the hidden Markov modeling approach because it is extensively used for time-varying phenomena. For speech recognition applications, an additional modeling procedure called Gaussian mixture modeling is applied to the framework of HMM because the observable events from the speech features are continuous and multi-dimensional in nature. Training and usage of the HMM model is explained as the baseline system makes use of it.

For any speech recognition to be of practical use, some form of noise robustness must be present in the system especially for environments with under 20 dB of signal-to-noise ratio present[1]. While there are many techniques in the field, those that were employed for the baseline system are those techniques that solves for training and testing mismatch, called feature compensation methods. These techniques are explained and discussed in detail, as they are also used for this thesis.

Finally, the chapter ends with a discussion of the current performance of the baseline system including an exposition of the problems that triggers the motivation for this thesis.

3.2 System Dimensions

Aside from the fact that the field of ASR is multi-disciplinary, focusing the attention to different possibilities for the target task makes it a very broad topic. At the onset, one can talk about the following different dimensions for a given task:

- **Target Language.** Different languages have different characteristics that can manifest in the speech signals. Tonal languages like Chinese and Thai require more sophisticated acoustic modeling while there are certain nuances available only to some specific language. Language models can also depend largely on the language used and specific techniques can be applied.
- **Recognition Task.** Tasks can range from isolated word recognition to keyword spotting to continuous speech. Even these choices can branch to different conditions for the task depending on the application. For example, techniques that apply to read speech may fail when used for spontaneous speech.

- **Vocabulary Size.** The words that need to be recognized can range from a few tens to more than 100,000 words.
- **Target Users.** Embedded in the concept of acoustic modelling is the choice of whether models are generated separately for specific categories such as age, gender, voice type, etc. A more challenging problem is that of speaker-independent modeling where a global model is used for any user of the system.
- **Expected Topic.** Depending on the task, developers may limit the models by considering the expected topic. Some can make more general models by allowing a degree of freedom for new words to be recognized.
- **Availability of Data.** A very important issue in developing ASR systems is that of whether the system is data-driven or not. A data-driven approach relies on the availability of training data for modeling. Recently, however, there is an increasing interest in developing methods for insufficient data.
- **Channel Conditions.** A natural consequence of having different tasks is the possibility of having many possible channel conditions. This includes, but is not limited to, acoustic environments, speech capturing devices, etc. This consideration led to the subfield of far-field ASR.

Despite these differences in application, most of the algorithms developed for ASR can be applied in general by considering a balance in processing complexity and accuracy. In fact, some research delves on the possibility of unifying the development of ASR by pointing to an all-specific speech recognition system.

For this thesis, the focus is on a modified feature extraction scheme. Feature extraction represents the block that provides the input to the speech recognition system and can be applied to a wide-array of speech processing applications, even outside ASR. For this reason, a basic low dimensional speech recognition system is appropriate to expose the inherent capabilities of the feature extraction scheme. State-of-the-art techniques for speech recognition, when applied simultaneously to a new approach could blur these inherent characteristics.

The speech recognition system that will be described for the rest of this chapter pertains to a Japanese language-based, isolated-word, 142 vocabulary, speaker-independent, commands-related, hands-free speech recognition system.

3.3 The Speech Recognition System

The architecture employed by the isolated-word speech recognition system is shown in Figure 3.1. It is divided into two phases, the upper portion is called the training stage and the lower portion, the recognition stage. The recognition stage is used for the actual usage of the system and for evaluation purposes, in which case the system is connected to some performance calculation module.

The training phase involves two major blocks, the feature extraction and the model estimation modules. The feature extraction is lumped into a set of processes called the front-end processing, which also involves data retrieval and storage for later use. Storage is included in the procedures because the training phase is regarded as an off-line procedure, and does not require real-time considerations. The model estimation block involves procedures pertaining to training of the acoustic model and storage as well.

The recognition phase involves data acquisition and feature extraction but differs from the training phase because the data are not stored. Rather, the data goes through a model comparison block, which compares the unknown input with the stored models. The nearest model is then declared as representative of the new, unknown input.

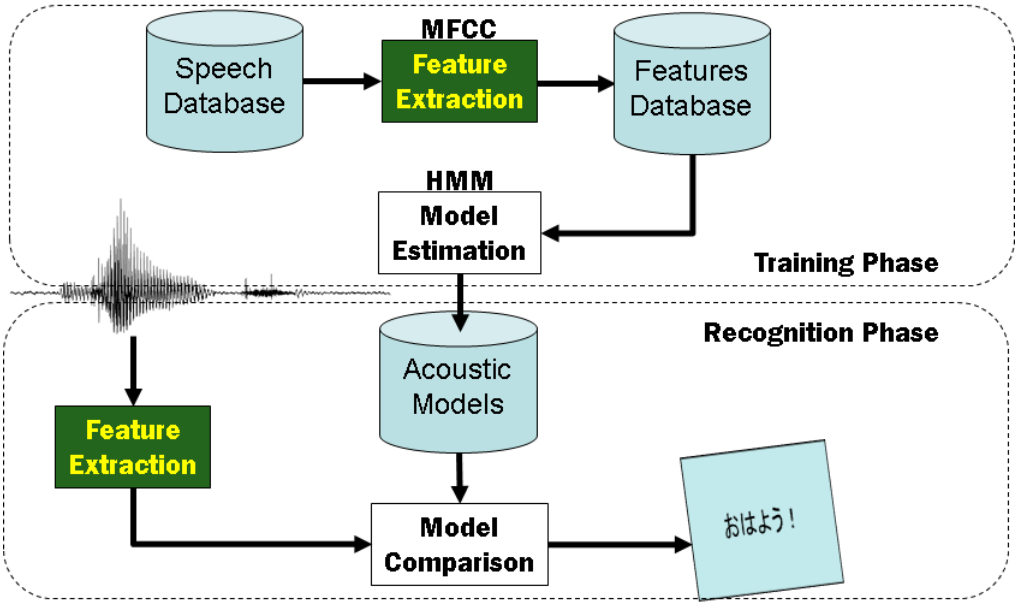

FIGURE 3.1: Isolated-word speech recognition system used for this thesis.

Many tools exist for implementation, simulation, and development of ASR systems. For this thesis, MATLAB is used as the simulation environment.

3.4 Front-End Processing

The front-end processing block is the main interface of the speech recognition system for acquiring and processing the speech input. In the acquisition phase is where sampling rates and channel condition decisions are made. Because of channel compensation techniques, the usually strict and specific interfacing requirements are relaxed. For example, a wide range of condenser microphones or even built-in microphones and headsets would work for the system as long as the proper sampling rates and SNR levels are met. Once the desired speech signal length is acquired, the process of feature extraction is done, which makes a compact representation of the speech signal that will be stored as a pattern in the acoustic modeling process.

Shown in Figure 3.2 are the three most widely used feature extraction methods based on the cepstral domain of the speech signal. Cepstral domain analysis allows for the concept of source separation, which in turn is based on a physiological understanding of the speech production mechanism. The basic definition of the cepstrum is that it is the inverse spectrum of the log spectrum of a signal. The reason for doing this is that the logarithm operation could compress the dynamic range of the spectrum and reduce amplitude differences in the harmonics. Treating the log spectrum as a waveform and performing an IFT leads to the cepstral domain. Source separation is achieved based on the fact that truncating the cepstral domain with increasing number of samples leads to increased spectral detail. This means that fast changing phenomena representing the source are positioned at higher samples in the cepstral domain.

The isolated-word speech recognition makes use of the Mel frequency cepstral coefficients or MFCC as its speech feature representation and will be described in detail. The perceptual linear prediction or PLP is a modification of the MFCC, as it includes additional auditory processing that may or may not be beneficial according to different results in the literature. The linear predictive coding or LPC is based on a source-filter model of the speech production mechanism, which

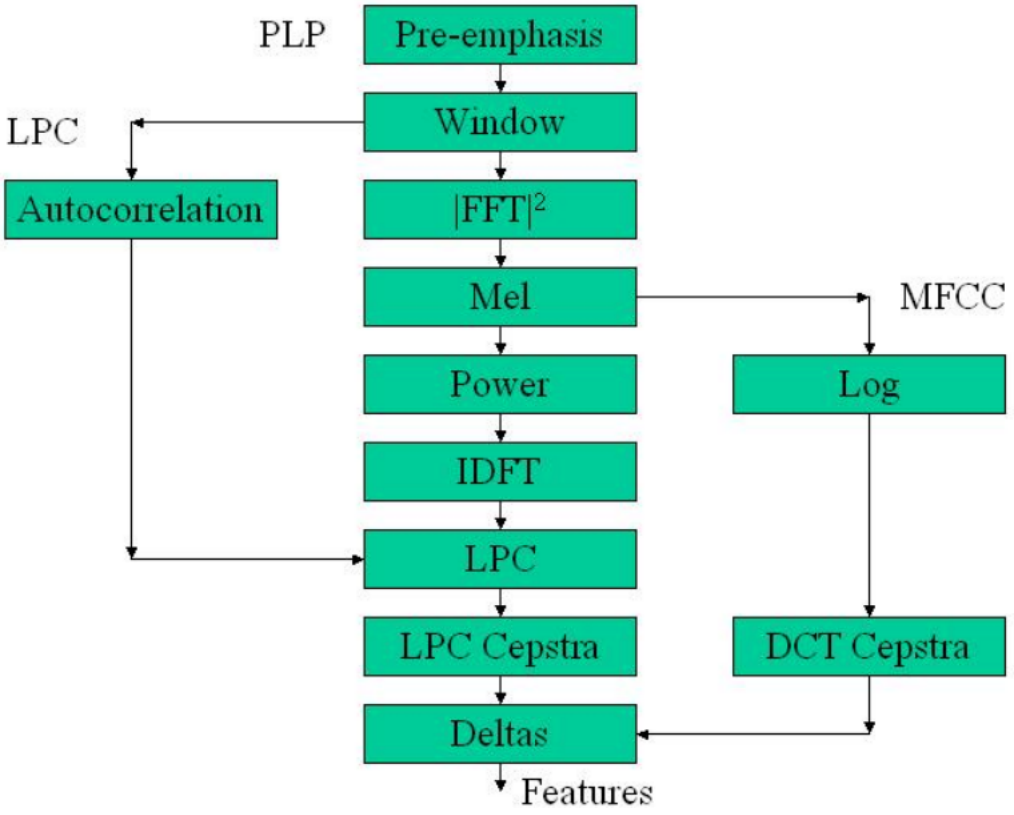

FIGURE 3.2: The three most widely used cepstral-based feature extraction techniques.

parameterize the speech spectrum as a set of filter coefficients. While the PLP feature extraction method is not used in this thesis, the LPC path will be discussed with some modifications in the succeeding chapter.

3.4.1 Pre-emphasis

In the study of speech production, it can be shown that the pressure waves that are acquired in the speech acquisition block is influenced by radiation impedance from the lips. This radiation gives a drop to the high-frequency components at about 6 dB per decade. The reason for pre-emphasis is to reverse this effect by approximating the radiation as a derivative, which can be modeled as an FIR filter with a single zero:

$$H(z) = 1 - \mu z^{-1} \tag{3.1}$$

where $\mu \in [0.95, 0.99]$. A common choice is to set $\mu = 0.97$ and is used all through out the experiments using pre-emphasis. This leads to the difference equation

where $y[n]$ is the pre-emphasized speech signal and $x[n]$ is the raw speech input:

$$y[n] = x[n] - 0.97x[n-1] \tag{3.2}$$

Aside from compensating for the high-frequency suppression, pre-emphasis also smooths the signal spectrum and allows for an unbiased frequency weighting when SNR level is computed for noise mixing.

3.4.2 Windowing and Power Spectrum Computation

As mentioned, the common practice for speech processing is to divide the signal into blocks before processing. The reason is not only because of real-time processing but also due to the nonstationary nature of the signal. Spectral content of speech changes over time depending on the spoken sound. These changes are not reflected when the processing is done to the entirety of the speech signal. As will be highlighted in the next chapter, choosing the length for the window can cause a time-frequency tradeoff. For most of the short-time feature extraction methods used in the field, the length of windows that are short enough to allow for the quasi-stationary or nearly stationary assumption is acceptable. This interval is at the range of 5 to 30 milliseconds, depending on the sampling rate and spectral content.

The process for short-time analysis used in the HU-SCS system is as follows:

1. Analysis window length of around 23.22 milliseconds is defined with a sampling rate of 11.025 kHz. This gives 256 samples for every frame.

2. To account for abrupt changes due to frame blocking, overlaps in between frames are defined. A 50% overlap is used for all experiments involving the use of overlaps. For this system, this is a shift length of 11.61 milliseconds or an overlap of N_0 =128 samples between adjacent frames.

3. A windowing function is chosen with the intention to smoothen the edges of every frame, which further avoids unnatural discontinuities in the analysis. The windowing function is selected by trading off the width of the main lobe and the attenuation of the side lobes in the spectral domain. A raised cosine

or Hamming window is used for this system:

$$w[n] = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right), \quad 0 \leq n \leq N-1 \tag{3.3}$$

4. The windowed segments $s_m[n]$ are generated by sliding the windowing function $w[n]$ on multiples of N_0 samples and multiplying it with the pre-emphasized speech signal $y[n]$:

$$s_m[n] = y[n + mN_0]w[n] \tag{3.4}$$

This is equivalent to sliding the signal on a stationary window starting from $n = 0$ at multiples of N_0.

5. Finally, apply FT analysis via FFT to each window segment $s_m[n]$:

$$S[m,k] = \sum_{n=0}^{N-1} s_m[n]e^{-j2\pi k/N}, \quad 0 < k < N_{\text{fft}} \tag{3.5}$$

where N_{fft} is the FFT width, which is set 512 samples. The power spectrum is then computed via:

$$|S[m,k]| = S[m,k]S^*[m,k] = |S[m,k]|^2 \tag{3.6}$$

3.4.3 Mel-Frequency Spectrum

The Mel part of the name of MFCC comes from a filter bank whose function is to group together certain bands according to the mel-scale, which is based on human perception. This mel-frequency scale is related to the linear frequency according to:

$$f_{\text{mel}} = 2595\log_{10}\left(1 + \frac{f}{700}\right) \tag{3.7}$$

which is linear up to around 1000 Hz and suddenly suppresses frequencies above it in a logarithmic fashion. Based on this characteristic, critical bands of the filter bank are designed as triangular filters $\Lambda_r[k]$ whose bandwidths are constant for center frequencies below 1000 Hz and increases at an exponential rate up to half the sampling rate as shown in Figure 3.3. Passing the DFT values $|S[m,k]|$ to the R filters, the rth value of the mel spectrum $M[r]$ with lower frequency L_r and

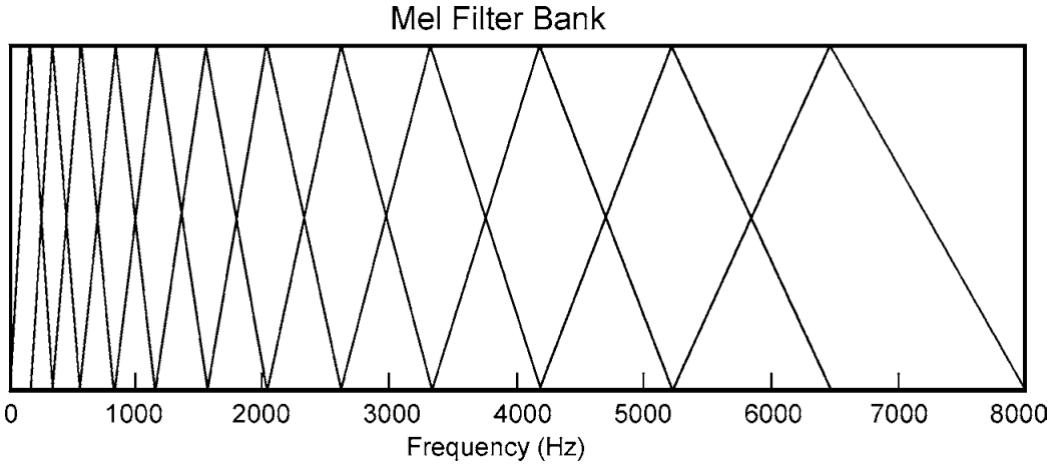

FIGURE 3.3: Triangular filterbank example for mel-frequency computation.

upper frequency U_r can be defined as:

$$M[r] = \frac{\sum_{k=L_r}^{U_r} |\Lambda_r[k] S[m,k]|}{\sum_{k=L_r}^{U_r} |\Lambda_r[k]|^2} \tag{3.8}$$

For the HU-SCS system, 13 linear filters were used below 1000 Hz and 37 filters up to half the sampling frequency for a total of R =40 filters.

3.4.4 DCT on the Log Spectrum

Finally, to get the MFCC samples, a discrete cosine transform is applied to the logarithm of the mel filterbank outputs. The discrete cosine transform is of the second type, which is more efficient than doing an IFT as it decorrelates the log energies of the mel-scale frequency:

$$C_m[k] = \frac{1}{R} \sum_{r=1}^{R} \log(M[r]) \cos\left[\frac{2\pi}{R}\left(r + \frac{1}{2}\right)k\right] \tag{3.9}$$

As mentioned, the goal of cepstral processing is to separate the source in order to get the slow changes of the vocal tract that defines the signal. This is done by getting M samples of the cepstrum, which is typically $M < R$. For this system,

the number of extracted cepstral samples is set to 12. For every frame, a set of 12 cepstral samples is considered as a feature vector.

The system also makes use of the log energy E_m:

$$E_m = \sum_{r=1}^{R} \log(M[r]) \tag{3.10}$$

to set the length of the feature vectors to 13.

3.4.5 Delta Cepstrum

Calculation of differentials between adjacent feature vectors has become a staple in speech recognition because it provides significant increase in accuracy. The reason for this will be explained more in the succeeding chapter. These time derivatives reflect the transitional changes happening in between feature vectors. Empirical results have shown that derivatives up to the second order can provide significant increase in performance. The deltas or first derivatives are computed as:

$$\Delta C_m[k] = \frac{\sum_{\tau=-T}^{T} C_{t+\tau}[k]}{\sum_{\tau=-T}^{T} \tau^2} \tag{3.11}$$

and the second derivatives as:

$$\Delta\Delta C_m[k] = \frac{\sum_{\tau'=-T'}^{T'} \Delta C_{t'+\tau'}[k]}{\sum_{\tau'=-T'}^{T'} \tau'^2} \tag{3.12}$$

The values of T and T' for this system was set to 2 and 1, respectively. The resulting values are then concatenated to the original 13-element feature vectors, which results to 39-element MFCC vectors that define the baseline system features.

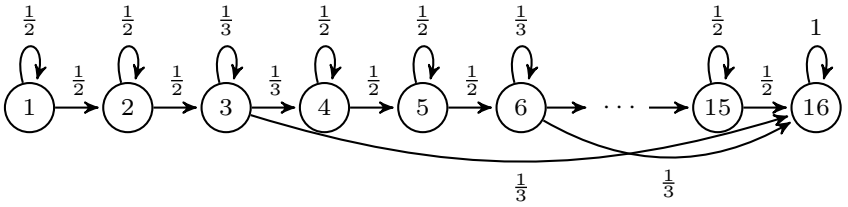

FIGURE 3.4: An example of a Bakis or left-to-right HMM topology with skipping.

3.5 Acoustic Modeling

The acoustic modeling block can be considered as the core mechanism that allows the possibility of pattern matching for the decoding process. The acoustic model makes a generalized inventory of the statistics of the acoustic units used in the system. Different studies in isolated-word speech recognition use different types of acoustic models, depending on the task complexity. For a medium size vocabulary task, the HMM modeling paradigm is well-suited. An understanding of HMM modeling is founded on the theory of discrete Markov processes, both using probabilistic transitions to model system events. In the latter, system states are assumed to be uniquely attached to one particular event and no information is hidden. Thus in DMPs, once an observation is made, the model state is easily deduced. For temporally complex events such as changes in speech feature vectors, a more complex modeling is required. This is the role of HMM modeling. In HMM, the observable events are now probabilistic functions of the model states. Therefore, to know the state of an observable event, the way to find out is to look at all the possibilities and decide based on a likelihood metric.

It should be noted, however, that the HMM does not have a single structural representation and the topology is dictated by empirical studies. For modeling events that change over time like speech, the usual topology employed is a left-to-right model or Bakis model as shown in Figure 3.4. Aside from the topological structure, the acoustic units that the model will represent is also application-dependent. Different task complexities call for different acoustic units. As an example, for a continuous speech recognition system with a 10,000-word vocabulary, 10,000 HMM models are not practical. Because of this, subword units such as diphones or phonemes were used as basic units since these are shared between words, generating a smaller number of acoustic models. For the system used for

this thesis, only 142 words are used and word units will not deal a great amount of storage and complexity problems. The HMM topology used is a 32-state left-to-right model without any skips.

3.5.1 HMM Training

In order to describe the training done for the system, what defines an HMM model is first described. HMMs are specified by two scalar values and three probability distributions. To facilitate a time-step analysis, variables are also defined. An event state at time t is set to a variable q_t. An observation at time t is set to the variable o_t. The scalar values of the HMM model are:

- N, the number of states in the hidden Markov chain. The states then could be defined as $S = \{S_1, S_2, \ldots, S_N\}$.
- M, the number of discrete observation symbols in every state. The symbols are defined as $V = \{v_1, v_2, \ldots, v_M\}$.

and the probability distributions:

- $A = \{a_{ij}\}$, the state transition probability. The probability of a transition from state S_i to state S_j is a_{ij}. Mathematically,

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i) \tag{3.13}$$

- $B = \{b_j(k)\}$, the observation probability distribution. The probability of emitting symbol v_k in state S_j is $b_j(k)$. For speech recognition, these emissions are in the form of vectors. Therefore, multinomial continuous distributions are used. Mathematically,

$$b_j(o) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(o, \mu_{jk}, \Sigma_{jk}) \tag{3.14}$$

  where the symbols V are replaced by an observation vector o, and $b_j(o)$ is defined by a mixture of M Gaussian components. This is the concept of GMM modeling for estimating probability densities as illustrated in Figure

3.5. Here, the elements of o has normal distributions defined by $o \sim \mathcal{N}(\mu, \sigma)$ and c is the scaling factor.

- π, the initial state distribution. The probability of starting the sequence at state S_j is π_j.

$$\pi_j = P(q_1 = S_j) \tag{3.15}$$

which are typically written as a triplet in compact form: $\lambda = \{A, B, \pi\}$.

The steps for training will be described for a single observation sequence:

1. Initialize the HMM model λ. A and π are usually initialized using uniform values, whereas B is set according to some predefined distribution.
2. Using Forward-Backward algorithm, compute for:
   - $\alpha_t(i)$, the probability of the training sequence up to time t and the state S_i at time t, given model λ:

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, q_t = S_i|\lambda) \tag{3.16}$$

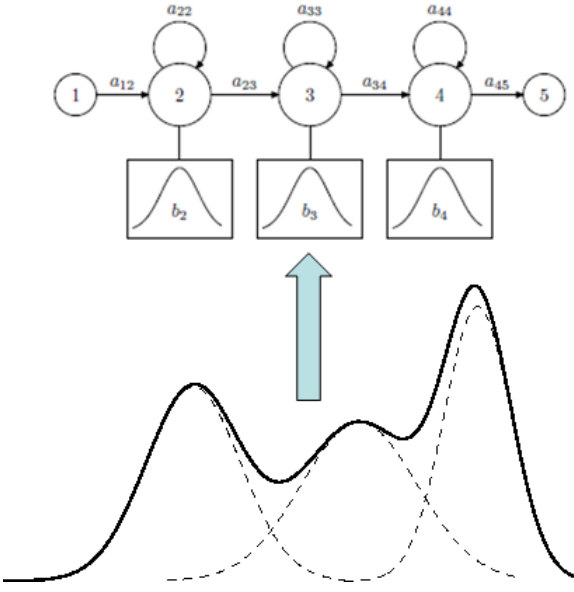

FIGURE 3.5: Illustration of Gaussian mixture modeling.

- $\beta_t(i)$, the probability of the partial observation sequence from $t+1$ up to the end T, given state S_i at time t and model λ:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T | q_T = S_i, \lambda) \tag{3.17}$$

3. Using the computed values $\alpha_t(i)$ and $\beta_t(i)$, compute for the following probability:

   - $\gamma_t(j,k)$, which represents the probability of being in state S_j at time t with k-mixture components accounting for observation o_t, given both the training sequence O and model λ:

$$\gamma_t(j,k) = P(q_t = S_j | O, \lambda) \tag{3.18}$$

   - $\xi_t(i,j)$, representing the probability of being in state S_i at time t, and S_j at time $t+1$:

$$\xi_t(i,j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \tag{3.19}$$

4. Finally, update the values of the model λ using the computed values:

   (a) For π_i, the expected frequency of being in state S_i at time 1:

$$\pi_i' = \gamma_1(i) \tag{3.20}$$

   (b) For a_{ij}, using the ratio of the expected frequency of state S_i to state S_j transitions over all the expected frequency of transitions from state S_i:

$$a_{ij}' = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.21}$$

   (c) For $b_j(o)$, the mixture components are updated individually. Starting with the scaling factor c_{jk}, using the ratio between the expected number of times the system is in state S_j using the k-th mixture component,

and the expected number of times the system is in state S_j:

$$c'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \tag{3.22}$$

the means μ_{jk} and variances Σ_{jk} are updated using:

$$\mu'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) o_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{3.23}$$

$$\Sigma'_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)(o_t - \mu_{jk})(o_t - \mu_{jk})^T}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{3.24}$$

This whole procedure of updating the parameters and iterating is the Baum-Welch algorithm.

5. The training procedure is done with the objective of updating the parameters λ to maximize $P(O|\lambda)$ with the maximum likelihood criterion. Using Forward-Backward algorithm, compute for this likelihood of the training sequence with the updated model λ^{new} by enumerating every possible state sequence and evaluating the corresponding probability:

$$P(O|\lambda^{\text{new}}) = \sum_{q_1, q_2, \ldots q_T} \pi'_{q_1} b'_{q_1}(o_{q_1}) a'_{q_1 q_2} b'_{q_2}(o_{q_2}) \ldots a'_{q_{T-1} q_T} b'_{q_T}(o_{q_T}) \tag{3.25}$$

Note that this computation involves many probabilistic values between 0 and 1 and rescaling has to be done to avoid underflows.

6. Finally, repeat the updating process until a local optimum for $P(O|\lambda)$ is found.

3.6 Decoding

The decoding process is coupled with the acoustic modeling algorithm used for the system. Among all the possible state sequence, find the single best state sequence path. Mathematically, this means finding the maximum $P(O|Q,\lambda)$ which is accomplished using the Viterbi algorithm. The procedure starts by defining a variable $\delta_t(i)$ representing the highest path probability containing t observations that ends at state S_i:

$$\delta_t(i) = \max_{q_1,q_2,\ldots q_{t-1}} P(q_1, q_2, \ldots, q_t = S_i, o_1, o_2, \ldots, o_t|\lambda) \tag{3.26}$$

Using induction, $\delta_{t+1}(j)$ can be computed as:

$$\delta_{t+1}(j) = \max_i \left[\delta_t(i)a_{ij}\right] b_j(o_{t+1}) \tag{3.27}$$

As the induction procedure is continued, the states being added to the path must be tracked by constructing an array:

$$\psi_{t+1}(j) = \arg\max_{1\leq i\leq N} \left[\delta_t(i)a_{ij}\right] \tag{3.28}$$

The Viterbi algorithm, can then be described by a step-by-step procedure:

1. Initialize $\delta_1(i)$ and $\psi_1(i)$ as:

$$\delta_1(i) = \pi_i b_i(o_1) \tag{3.29}$$

$$\psi_1(i) = 0 \tag{3.30}$$

   for all values of $1 \leq i \leq N$.

2. For $2 \leq t \leq T$ and $1 \leq j \leq N$,recursively compute for:

$$\delta_t(j) = \max_{1\leq i\leq N} \left[\delta_{t-1}(i)a_{ij}\right] b_j(o_t) \tag{3.31}$$

$$\psi_t(j) = \arg\max_{1\leq i\leq N} \left[\delta_{t-1}(i)a_{ij}\right] \tag{3.32}$$

3. For the last sequence, compute for:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)] \tag{3.33}$$

$$q_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)] \tag{3.34}$$

4. Finally, the best-state sequence is retrieved by backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \tag{3.35}$$

3.7 Noise Compensation Techniques

3.7.1 Cepstral Mean Subtraction

Cepstral mean subtraction or CMS is a channel normalization approach to compensate for the acoustic channel. The time invariant channel parameters in a recording system and convolutional disturbance noise are evaluated by CMS and these noises are reduced from an observed speech waveform. By using CMS, the distortion between the training speech data and the observed speech data can be improved. If $C_m(k)$ represents the feature vector at time m, then the CMS-applied feature vector $\tilde{C}_m(k)$ is defined as:

$$\tilde{C}_m(k) = C_m(k) - \frac{1}{T} \sum_{\tau=1}^{T} C_\tau(k) \tag{3.36}$$

In CMS, the averages of all the feature components are calculated per utterance, and the averages are subtracted from the values themselves. This is similar to z-score normalization which involves CMN and CVN. CMS can eliminate channel effects caused by multiplicative noise such as microphone mismatch and distortion caused by the transmission channel. The multiplicative factor becomes an addition operation after the logarithm scaling of the cepstrum. The process also requires no estimation procedure except for the averages.

3.7.2 Filtering in the Modulation Spectrum

If the spectrum obtained in the speech analysis is taken as a time-series data in frequency, the resulting spectrum is called the modulation spectrum. In this modulation spectrum, the important components of speech is said to be present in the range of 1 to 10 Hz. On the other hand, the influence of multiplicative noise is concentrated below 1 Hz. Therefore, by using a high-pass filter with a cut-off at 1 Hz, we can eliminate some parts of the multiplicative noise. Some studies also used bandpass filters from 1 to 12 Hz.

By using an IIR filter, there is a possibility that the resulting filter could be unstable due to limit cycles caused by quantizations leading to cancellations of poles or zeros. Therefore, the use of stable FIR filters have been proposed for the baseline system. This technique is called running spectrum filtering or RSF in order to differentiate it from RASTA.

3.7.3 Dynamic Range Adjustment

Noise that cannot be solved by CMS is addressed by DRA. These can be perturbations caused by the other enhancement methods. For example, filtering in the modulation frequency, which may also affect desired data. Therefore, this process is usually undertaken as the last noise compensation process. If $C_m(k)$ represents the feature vector at time m, then the DRA-applied feature vector $\tilde{C}_m(k)$ is defined as:

$$\tilde{C}_m(k) = \frac{C_m(k)}{\max\limits_{\tau=1,\ldots,T} C_\tau(k)} \tag{3.37}$$

3.8 Problem in Baseline System

To conclude this chapter, the confusion matrix of the best-performing system under clean channel conditions is shown in Figure 3.6. Using numerical analysis, it was found that around 6% of the errors generated by the best-performing system comes from a clear confusion in between words that are near in pronunciation. The confusion matrix clearly shows that there are overlaps in false negative results in

between words that only differ by one or two phonemes. The next chapter will discuss how this thesis proposes to solve this problem.

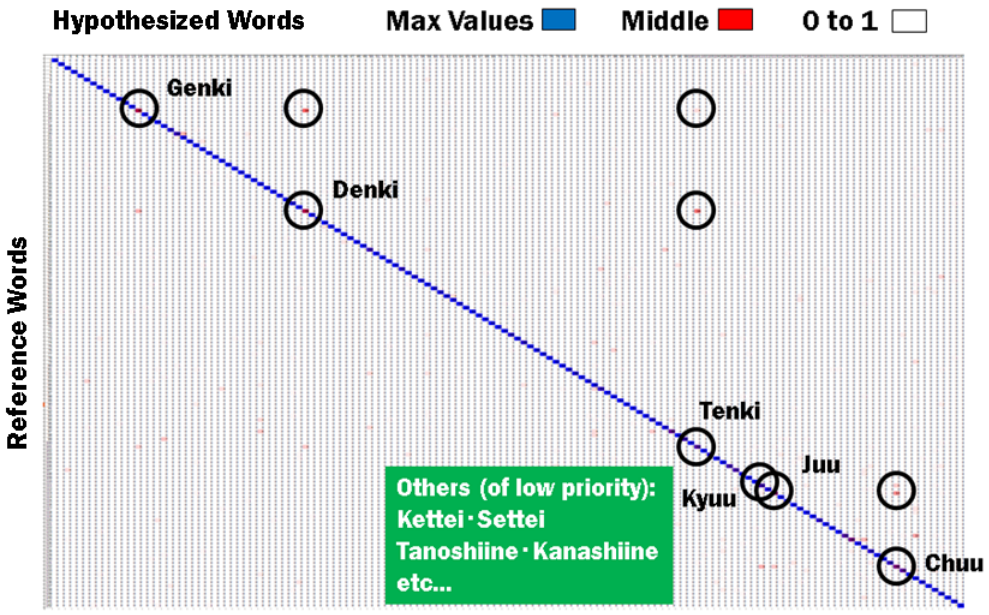

FIGURE 3.6: Baseline system confusion matrix

Chapter 4

Feature Extraction Modifications

4.1 Chapter Overview

As concluded in the previous chapter, some problems have yet to be solved in developing speech recognition systems. The analysis of the confusion matrices have clearly shown that majority of prevalent errors in the current system are due to words that are almost similar except for a transient part. This chapter expounds on the details of the limitations of the standard speech feature representation that is used in the previous system and how it can be addressed.

The techniques for feature extraction presented in the previous chapter such as the MFCC, PLP, and LPCC were not only driven by studies in speech signal processing but also by available technology, processing resources and real-time application requirements. Because of this, short-time framing of signals for processing became a staple in most development for speech signals. While this concern has been acknowledged even since the 1950's, developments did not meet much success due to the underlying complexity of improving time-frequency resolution. For example, bilinear time-frequency distributions is one step above conventional analysis as it aims to accurately represent the energy of a signal in both time and frequency domains.

For the case of ASR, its development continued with the stationary feature extraction methods, only to be augmented with dynamic information to reflect transitional changes. Even for the case of continuous speech, acoustic modeling only

accounts for a share of the overall accuracy and the recognition system has to rely on other sources of information such as language models and adaptive techniques in order to improve. These augmentations suggest that techniques that would allow the rich, dynamic information to be reflected in the model could aid systems that rely on more accurate speech models. Thus, efforts in nonstationary or time-varying signal processing research are being conducted.

There exists a number of techniques suitable for ASR such as diphone-based modeling, AM-FM modeling. However, the most studied class is that of parametric speech models that assumes AR or ARMA models for synthesis. Solving for the parameters of these models has been well-studied and the common time-varying extension is to allow these parameters to be time-varying[26, 27]. There are said to be roughly three classes of models in the literature, the differences of which are based on how the prediction coefficients are defined:

1. adaptive models [28]
2. explicit basis function models
3. random (mostly Markovian) models [29]

For the case of speech recognition, the use of a linear combination of basis functions seems to have a good compromise between the aforementioned requirements. This then became the topic of a number of research, giving a basic formulation of the extension and a discussion of issues involved. However, only a few have attempted to apply it in actual speech recognition systems due to issues that are explained in this chapter. The complete procedure employed for this thesis is detailed in the rest of the discussion, and will conclude with the proposed models.

4.2 Short-Time Speech Feature Representation

Breakthroughs in speech signal processing are founded in a continuous development that also involved considerations in technology and real-time processing. As a practical example, a professional English speaker can talk fast at an average rate of 160 words per minute. The said rate translates to one word every 0.375 second. Using narrowband processing, which is the lower limit for conversational speech,

the average word is digitally processed at a rate of 8,000 samples every second. At this rate every word will produce an average of 3,000 samples, which amounts to 3,000 numerical values that have to be stored and processed. For a simple radix-2 fast Fourier transform operation, a word signal that has to be analyzed leads to 49,152 operations. The latency of this process depends on the duration between the first and last samples in the word signal, and the time it takes to finish the operation. Processing also relies on the available technology and so, limitations have to be imposed in order to satisfy the memory and real-time requirements of the target speech communication system. For the case of speech processing, this process of dividing the signal into smaller frames of samples also facilitates batch processing of incoming signals. This block processing of speech as input is common in speech recognition systems.

The speech signal, the acoustic waveform that a speaking person produces, contains information about the air pressure that is released from the lungs, and undergoes modifications depending on how the vocal tract is moved at any instant. Due to differences in physiology and articulation from one person to another, or even with the same person, the signals representing multiple utterances of the same word will have differences, however minor. For speech recognition, system performance depends on a resource-efficient detection of (1) differences between different words and (2) similarity between multiple instances of the same word despite the expected differences. This leads to the notion of finding a compact representation of speech that helps in achieving the aforementioned goals through the extraction of relevant speech parameters. This is more commonly known as feature extraction in the field of speech recognition.

In addition to the suboptimal characteristics of the speech input, the environment and external factors can also contribute to the occurring signal variations. Thus, robustness against variations and noise is also a major concern in developing speech features. Psychoacoustics and physiological studies have allowed researchers to set boundaries for the model space of speech signals. These studies have led to reductions in the sensitivity of the speech features to speaker variations. However, these models were developed with short-time or block-based analysis in mind.

While short-time analysis is considered as a principal tool of the speech community, it also presents an impediment to speech features as used for speech recognition. This is because while the speech signal is considered as a time-varying

phenomenon, reliance to short-time analysis has led to models that require stationarity within each frame. Hence, deciding on the constant length of each frame creates suboptimality for the representation. Deciding on this length of the analysis window has the following implications:

1. There is a time-frequency resolution trade-off that favors time when the window length is short, and vice versa.
2. If the window length is in the order of the pitch period for a voiced sound, variations in analysis will occur depending on the location of the analysis window within the pitch period. As window length is increased, the variation is also diminished.
3. Transient unvoiced sounds are increasingly blurred by longer windows.
4. Bias-variance trade-offs for additive noise and variations caused by amplitude modulation.

Despite these considerations, studies have allowed for empirical choices for the frame length based on the concept of quasi-stationarity of speech signals. In general, this is in the order of 10 to 30 milliseconds of speech per frame. While the choice was proven to be effective when the system performance is compared to random guessing, the short-time analysis window still violates the quasi-stationary assumption. This is especially evident in transient parts of the speech signal, which makes the model highly dependent on the correct representation of voiced sounds. As a result, the speech recognition system becomes highly reliant on vowel recognition.

In Furui[30] and Jenkins[31], it was shown that humans perceive VC and CV transitions as larger sources of information for correctly identifying phonetic contexts, rather than the longer stationary portions. This led to the use of differential spectral parameters, which captures the relative changes or transitions between frames. These became more commonly known as delta and delta-delta parameters for the first and second differentials, respectively. Incorporating these parameters, typically by concatenating with the initial feature vectors, improved the performance of speech recognition systems. Thus, developing a working theory for the transient and highly nonstationary parts of speech is a key area that can be explored to improve the overall recognition performance.

4.3 Time-Varying Linear Prediction

4.3.1 General Concepts and Scope

In the speech processing literature, different methods have been proposed to incorporate the nonstationary information of the signal. However, the techniques assume that the acoustic modeling is dependent on the modification that will be done to the feature extraction process. Given the nature of short-time analysis, which assumes that the signals within each frame interval are stationary, the extent of an automatic accommodation of non-stationarity was done by allowing speech model parameters to be time-dependent. This is the concept of time-varying analysis based on a parametric speech model.

The most widespread choice of parameterization in the speech processing literature for time-varying analysis are the AR and ARMA models[32]. The stationary AR model for synthesis as shown in Figure 4.1 assumes an all-pole digital filter model $H(z)$ with gain G and filter coefficients a_i with P values indicating the order:

$$H(z) = \frac{G}{1 + \sum_{i=1}^{P} a_i z^{-i}} \tag{4.1}$$

The simplification of the AR model from the general pole-zero transfer function of the ARMA is properly justified in the literature except for nasal sounds. Based on the complexity of the two models, the AR model is sufficient as it provides performance improvements at a lower cost.

For Equation 4.1, given an input $u[n]$ and output $s[n]$, the difference equation can be derived as:

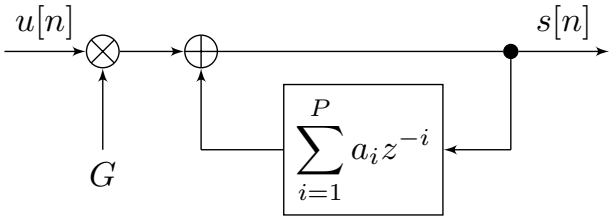

FIGURE 4.1: AR model for speech synthesis.

$$s[n] = -\sum_{i=1}^{P} a_i s[n-i] + Gu[n] \tag{4.2}$$

where $u[n]$ represents the actuating signal and $s[n]$ the generated speech signal. By making the assumption that the speech signal is a correlated Gaussian process, an approximate linear prediction can be made to a speech sample at any time using past samples:

$$\hat{s}[n] = -\sum_{i=1}^{P} \alpha_i s[n-i] \tag{4.3}$$

This change in notation from a to α also serves to reflect the fact that the actual speech signals will not be exactly equivalent to this model. To reflect the non-stationary nature of the speech signal $s[n]$, the parameters α_i are allowed to be time-varying:

$$\hat{s}[n] = -\sum_{i=1}^{P} \alpha_i[n] s[n-i] \tag{4.4}$$

Solving for the coefficients $\alpha_i[n]$ will pose a problem since there will be an infinite number of solutions and no inherent structure will show. The coefficients must then be constrained such that the resulting $\alpha_i[n]$ follows a time series decomposition in which the innovations have normalized and constant variance. These coefficients can be either solved adaptively if the time dependency is not explicitly defined or systematically if otherwise. The time dependency can be defined as a known stochastic process, or as a linear combination of known time functions.

4.3.2 Estimating TV-LPC Coefficients

For speech recognition, it was previously shown that a balance between accurate representation of the speech signal and parsimonious use of processing resources is necessary especially if the target is for real-time application that may involve dedicated hardware. With these considerations, constraining the coefficients $\alpha_i[n]$ as a linear combination of known time functions is a reasonable choice. Following Equation 4.4, $\alpha_i[n]$ can be written as:

$$\alpha_i[n] = \sum_{j=0}^{Q} A_{ij} f_j[n] \tag{4.5}$$

where $f_j[n]$ is the set of functions where several choices exist. Prior to 1986, the basis functions revolved around a few known functions chosen to reflect the overall smooth transition of the speech signal, notwithstanding the case of plosive sounds. During that time, the most common functions were those of powers of time, Fourier series, prolate spheroidal wave functions, and Legendre polynomials. Empirical results have shown similar results for these choices according to a comprehensive survey in Grenier [33, 34]. However, a common finding that is present in the use of these models is that they cannot ensure stability when least squares method of estimation is used. In Grenier the time-varying aspect is later extended to lattice structures employing parameters such as PARCOR coefficients and log area ratios of reflection coefficients in order to ensure stability[35]. Ensuring stability for the basic least squares solution would later become a research theme where attempts have led to alternative optimization metrics[27, 36–40]. In this thesis, these issues are resolved using the basic solutions as it focuses on a different aspect of utilizing the time-varying coefficients.

The basic least squares estimation method adopted in Liporace [41], Grenier[33, 34], and Hall[42] is derived based on the same idea for stationary LPC. Combining Equations 4.4 and 4.5, the prediction equation becomes:

$$\hat{s}[n] = -\sum_{i=1}^{P}\sum_{j=0}^{Q} A_{ij} f_j[n] s[n-i] \tag{4.6}$$

where as usual, the prediction error is defined as:

$$e[n] = s[n] - \hat{s}[n] \tag{4.7}$$

The optimization criterion for the least squared estimation method is the sum of the squared prediction error function:

$$\begin{aligned} J(A_{ij}) &= \sum_{n} e^2[n] \\ &= \sum_{n} \left(s[n] + \sum_{i=1}^{P}\sum_{j=0}^{Q} A_{ij} f_j[n] s[n-i] \right)^2 \end{aligned} \tag{4.8}$$

The objective then is to minimize J in order to get the optimal basis function coefficients A^*_{ij}:

$$A^*_{ij} = \arg\min_{A_{ij}} J(A_{ij}) \tag{4.9}$$

which in turn, allows for the solution of α_i according to Equation 4.5.

4.3.3 Solving TV-LPC Coefficients

Following the established theory for solving stationary LPC coefficients, the solution for time-varying LPC coefficients requires some preliminary assumptions. For example in Equation 4.8, no clear range for n is given to the objective function $J(A_{ij})$. The range can be either finite or infinite, which corresponds to the covariance and autocorrelation methods for the stationary case, respectively. While more efficient methods exist for the autocorrelation method when applied to time-invariant LPC, it was established in Hall[42] that for the case of time-varying LPC, the covariance method is more advantageous for the following reasons:

1. Distortions caused by discontinuities at both ends of the data interval that is present in the time-invariant autocorrelation method is also present for the time-varying case. For the covariance method, this is not true because the error minimization is only applied over the range where the P-order prediction can be applied.

2. A solution for reducing the distortions caused by the discontinuities is to use a windowing function. However, this causes more distortions for the time-varying LPC, as it is very sensitive to slight changes in the signal.

3. The autocorrelation method assumes that the signal is stationary over the time interval, contradicting the basic assumption of the time-varying LPC.

Following the results of deriving the objective function based on a least squares estimation method, Equation 4.9 can be expanded by working out the minimization

for $J(A_{ij})$ in Equation 4.8:

$$\frac{\partial J(A_{ij})}{\partial A_{pq}} = 0$$

$$2\sum_{n}\left(s[n] + \sum_{i=1}^{P}\sum_{j=0}^{Q} A_{ij}^{*} f_j[n]s[n-i]\right) f_q[n]s[n-p] = 0 \tag{4.10}$$

where $1 \leq p \leq P$ and $0 \leq q \leq Q$.

The grouping and order of Equation 4.10 can be rearranged such that the set of linear equations for solving A_{ij} becomes apparent:

$$\sum_{i=1}^{P}\sum_{j=0}^{Q} A_{ij}^{*}\left(\sum_{n} f_j[n]f_q[n]s[n-i]s[n-p]\right) = -\sum_{n} f_q[n]s[n]s[n-p] \tag{4.11}$$

where $1 \leq p \leq P$ and $0 \leq q \leq Q$.

A generalized correlation function can then be defined:

$$C_{jq}(i,p) = \sum_{n} f_j[n]f_q[n]s[n-i]s[n-p] \tag{4.12}$$

such that Equation 4.11 can be written as:

$$\sum_{i=1}^{P}\sum_{j=0}^{Q} A_{ij}^{*} C_{jq}(i,p) = -C_{0q}(0,p) \tag{4.13}$$

where $1 \leq p \leq P$ and $0 \leq q \leq Q$.

This can be further expressed in matrix form by defining the vectors

$$\mathbf{A}_j^T = \begin{bmatrix} A_{1j} & A_{2j} & \dots & A_{Pj} \end{bmatrix}, \quad 0 \leq j \leq Q \tag{4.14}$$

and

$$\mathbf{\Psi}_j^T = \begin{bmatrix} C_{0j}(0,1) & C_{0j}(0,2) & \dots & C_{0j}(0,P) \end{bmatrix}, \quad 0 \leq j \leq Q \tag{4.15}$$

and the matrix

$$\mathbf{\Phi}_{jq} = \begin{bmatrix} C_{jq}(1,1) & C_{jq}(1,2) & \cdots & C_{jq}(1,P) \\ C_{jq}(2,1) & C_{jq}(2,2) & \cdots & C_{jq}(2,P) \\ \vdots & \vdots & \ddots & \vdots \\ C_{jq}(P,1) & C_{jq}(P,2) & \cdots & C_{jq}(P,P) \end{bmatrix}, \quad 0 \leq j,q \leq Q \tag{4.16}$$

making Equation 4.13 compactly represented as:

$$\begin{bmatrix} \mathbf{\Phi}_{00} & \mathbf{\Phi}_{01} & \cdots & \mathbf{\Phi}_{0Q} \\ \mathbf{\Phi}_{10} & \mathbf{\Phi}_{11} & \cdots & \mathbf{\Phi}_{1Q} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Phi}_{Q0} & \mathbf{\Phi}_{Q1} & \cdots & \mathbf{\Phi}_{QQ} \end{bmatrix} \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_Q \end{bmatrix} = - \begin{bmatrix} \mathbf{\Psi}_0 \\ \mathbf{\Psi}_1 \\ \vdots \\ \mathbf{\Psi}_Q \end{bmatrix} \tag{4.17}$$

This final matrix form can also be rewritten such that the vectors and matrices are defined over the range of P, with the range of Q explicitly specified.

4.3.4 A Covariance Method Algorithm

The resulting matrix form of Equation 4.17 can be solved using different methods depending on the underlying symmetry. For the derived equations, it can be deduced from Equation 4.13 that the elements $C_{jq}(i,p)$, $C_{qj}(i,p)$, $C_{jq}(p,i)$, and $C_{jq}(p,i)$ will have equal values. This will mean that the matrices $\mathbf{\Phi}_{jq}$ in equation 4.17 are themselves symmetric and also forms block symmetry. In addition, depending on the chosen functions $f_j[n]$, further reductions in computations can be achieved.

An efficient and general method that can be used for solving the equations is the Cholesky decomposition, otherwise known as the square-root method, which is extensively used for time-invariant LPC. The formulation for the time-varying LPC is very similar to the time-invariant case. The subsequent formulas arising from defining the predictor coefficients as a linear combination of basis functions only differs from the time-invariant case in such a way that the number of scalars involved is increased by the order of the basis functions used. With the assumption that the derived result of Equation 4.17 can be solved as if it is the result of the invariant LPC formulation, the original algorithm can be used to solve for the

coefficients. First Equation 4.17 can be expressed as:

$$\mathbf{\Phi A} = -\mathbf{\Psi} \tag{4.18}$$

The first step is to express the matrix $\mathbf{\Phi}$ as

$$\mathbf{\Phi} = \mathbf{VDV}^T \tag{4.19}$$

where $\mathbf{V}$ is a lower triangular matrix whose main diagonal is composed of unity elements, and $\mathbf{D}$ is a diagonal matrix. Element-wise, these values are expressed as:

$$\phi(k,\ell) = \sum_{m=1}^{\ell} V_{km} d_m V_{\ell m}, \quad 1 \leq \ell < k \tag{4.20}$$

using algebraic manipulation:

$$V_{k\ell} d_\ell = \phi(k,\ell) - \sum_{m=1}^{\ell-1} V_{km} d_m V_{\ell m}, \quad 1 \leq \ell < k \tag{4.21}$$

and diagonal elements:

$$\phi(k,\ell) = \sum_{m=1}^{k} V_{km} d_m V_{km} \tag{4.22}$$

again, using algebraic manipulation:

$$d_k = \phi(k,k) - \sum_{m=1}^{k-1} V_{km}^2 d_m, \quad k \geq 2 \tag{4.23}$$

To solve for the elements, the first diagonal entry d_1 is initialized as:

$$d_1 = \phi(1,1) \tag{4.24}$$

and iteratively solve for each element using Equations 4.21 and 4.23. Once the all entries are solved, the linear equations can be viewed now as:

$$\mathbf{VDV}^T\mathbf{A} = -\mathbf{\Psi} \tag{4.25}$$

and the values of $\mathbf{A}$ can be solved by expressing 4.25 in terms of a dummy variable:

$$\mathbf{Y} = \mathbf{DV}^T\mathbf{A} \tag{4.26}$$

using a two-step procedure. The dummy variable can be recursively solved as:

$$Y_k = -\psi_k - \sum_{\ell=1}^{k-1} V_{k\ell} Y_\ell, \quad 2 \leq k \leq P(Q+1) \tag{4.27}$$

by initializing

$$Y_1 = -\psi_1 \tag{4.28}$$

After solving for all entries of $\mathbf{Y}$, the second step is derived by algebraic manipulation of the element-wise Equation 4.26:

$$a_k = \frac{Y_k}{d_k} - \sum_{\ell=k+1}^{P(Q+1)} V_{\ell k} a_\ell, \quad 1 \leq k < P(Q+1) \tag{4.29}$$

using the initial condition:

$$a_{P(Q+1)} = \frac{Y_{P(Q+1)}}{d_{P(Q+1)}} \tag{4.30}$$

4.4 Time-Varying Cepstral Coefficients

As mentioned, parametric models are used for representing speech because of the balance in representation and resource. However, another important aspect of speech feature representations is its robustness against variability. For speech recognition, it has been shown that features derived from vocal tract modeling and source separation are significantly robust against noise and variability [2, 3]. Specifically, while LPC coefficients can be utilized to be noise-robust [43, 44], they are not found to be effective as feature parameters [45]. For this reason, the solved coefficients from the linear prediction process outlined are further subject to cepstral conversion.

The nearest cepstral conversion for linear prediction coefficients is the one derived in Atal[46], called LPCC. A unique and simple relationship was shown to exist between the coefficients of the LPC synthesis filter and the time series samples of the logarithmic response. This can be derived by directly equating the logarithm of the filter transfer function and its power series expansion:

$$\ln\left(\frac{1}{1+\sum_{i=1}^{P} a_i z^{-i}}\right) = \sum_{n=1}^{\infty} c_n z^{-n} \tag{4.31}$$

Taking the derivative of both sides with respect to z^{-1}:

$$\frac{-\sum_{i=1}^{P} i a_i z^{-i+1}}{1+\sum_{i=1}^{P} a_i z^{-i}} = \sum_{n=1}^{\infty} n c_n z^{-n+1} \tag{4.32}$$

the denominator is multiplied to both sides, yielding:

$$-\sum_{i=1}^{P} i a_i z^{-i+1} = \left(1+\sum_{i=1}^{P} a_i z^{-i}\right) \sum_{n=1}^{\infty} n c_n z^{-n+1} \tag{4.33}$$

and a recursive relationship can be found by equating the constant term and various powers of z^{-1}:

$$c_n = \begin{cases} -a_1, & n = 1 \\ -a_n - \sum_{i=1}^{n-1} \left(\frac{i}{n}\right) c_i a_{n-i}, & 1 < n \leq P \\ -\sum_{i=1}^{n-1} \left(\frac{i}{n}\right) c_i a_{n-i}, & n > P \end{cases} \tag{4.34}$$

The recursion allows an infinite number of cepstral samples to be solved for one set of filter coefficients. These cepstral samples are different from the cepstrum derived directly from the speech signals as the trunction of a signal for Fourier analysis results to a time series with all zeros. The derived cepstral samples are directly taken from an all-pole transfer function. The recursive derivation also requires that the filter is stable since the sample represent the contributions of the poles scaled by the corresponding residue of each pole.

4.4.1 Stability Issues

A recurring observation in early studies of time-varying LPC based on the least square estimation method is that it is not guaranteed to be stable. Sets of filter coefficients result to sudden bursts in computed spectral estimates. While still considered to be smooth based on the temporal movement of the spectral estimate, the values of the coefficients are too high to be useful for conversion. Thus, in order

to convert LPC coefficients to their corresponding cepstra, stability of the resulting filters must be ensured. The first step for ensuring stability is by detecting unstable and marginally stable set of coefficients using a step-down procedure based on the lattice filter theory.

Given the Pth-order polynomial of the synthesis filter $A_P(z)$, the last coefficient is checked if it is greater than or equal to 1 as it represents the product of all the poles in the filter. This means that if,

$$A_P(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{P-1} z^{-(P-1)} + a_P z^{-P} \tag{4.35}$$

the Pth reflection coefficient is set as the last coefficient of the polynomial:

$$k_P = a_P \tag{4.36}$$

If $|k_P| \geq 1$, then one or all of the poles could be outside or marginally at the unit circle such as the one depicted in Figure 4.2. Otherwise, a recursive step-down process is done by solving for the lower-order filter $A_{P-1}(z)$:

$$A_{P-1}(z) = \frac{A_P(z) - z^{-P} k_P A_P(1/z)}{1 - k_P^2} \tag{4.37}$$

The $(P-1)$th reflection coefficient is again set and checked for $|k_{P-1}| \geq 1$. If the filter is stable, the process will continue until k_1 is reached. It should be mentioned that this process is equivalent to computing for the factorized form and searching for a pole that is greater than or equal to unity albeit more processor-intensive.

While it is possible to replace the unstable filter with an equivalent minimum phase, stable model using Schur factorization, the process did not provide significant improvements over the actual procedures employed. Thus, a simpler solution was used for dealing with unstable and marginally stable coefficients. For unstable filters, poles outside the unit circle are reflected inside the unit circle. For marginally stable filters, the previous coefficient vector is simply replicated.

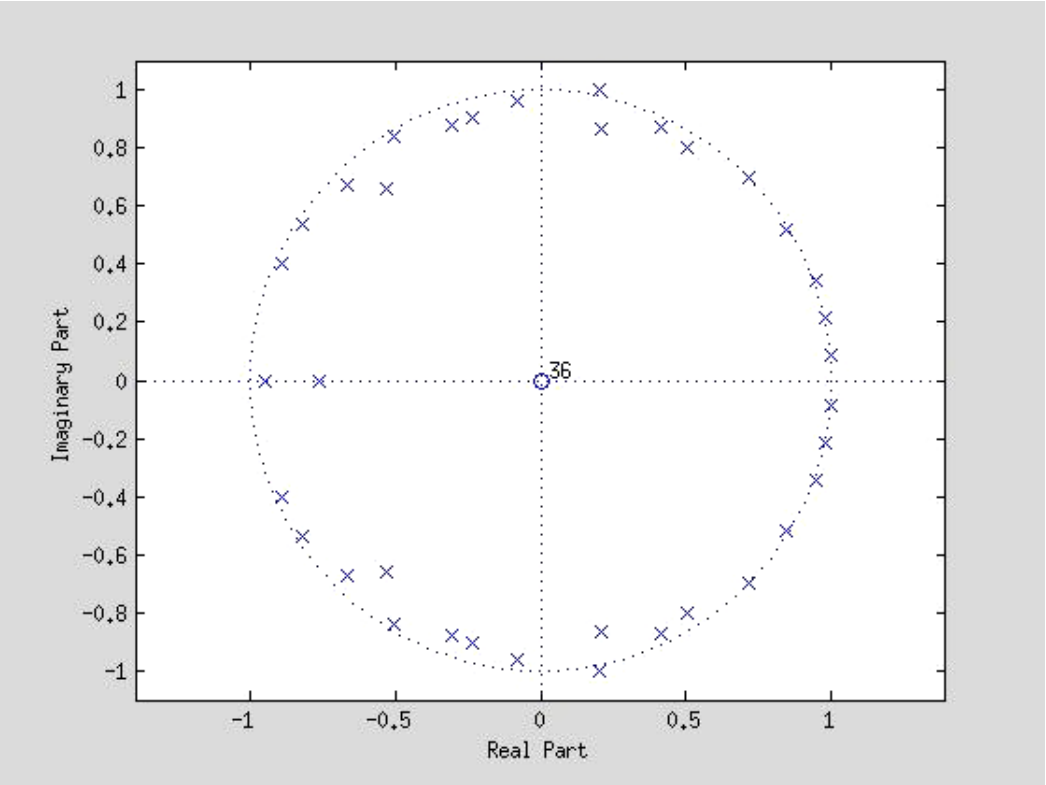

FIGURE 4.2: Pole-zero diagram of an unstable filter.

4.5 TV-LPCC Feature Extraction

The application of TV-LPCC coefficients is not as straightforward as it is expected to be because of a number of involved parameters and dependency on the target speech recognition architecture. At the onset, sensitivity concerns call for an efficient VAD system for consistent matching of signal end points. For the actual model itself, the order P of the estimation function and, the number of functions Q used for constraining the coefficients, and the basis functions themselves must also be decided. Short-time analysis requirements are still present although at a lower concern because the stationary assumption is relaxed. However, in terms of complexity and for evaluation and comparative purposes, a number of limiting assumptions still has to be done.

4.5.1 A Running Example

To put things in perspective, a running example using a speech signal sample will be used to clarify important details. The speech signal shown in Figure 4.3 is

subject to TV-LPCC feature extraction. This signal comes from the onset of the word *genki* in Japanese. Figure 4.4 shows the short-time log magnitude spectrum

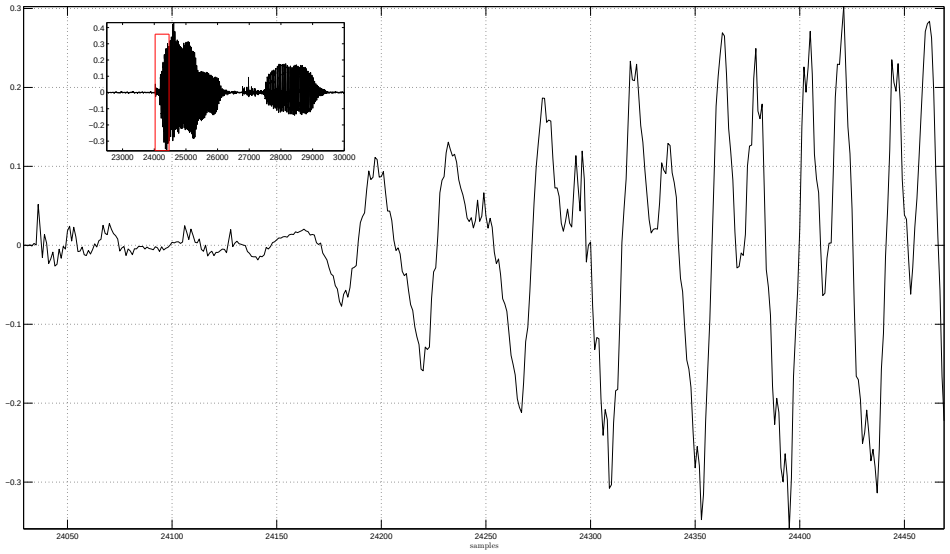

FIGURE 4.3: Speech signal sample for TV-LPCC extraction.

as computed by the FFT at 512 samples, and the corresponding spectral estimate based on time invariant LPC. Figure 4.5 then shows a possible set of spectral

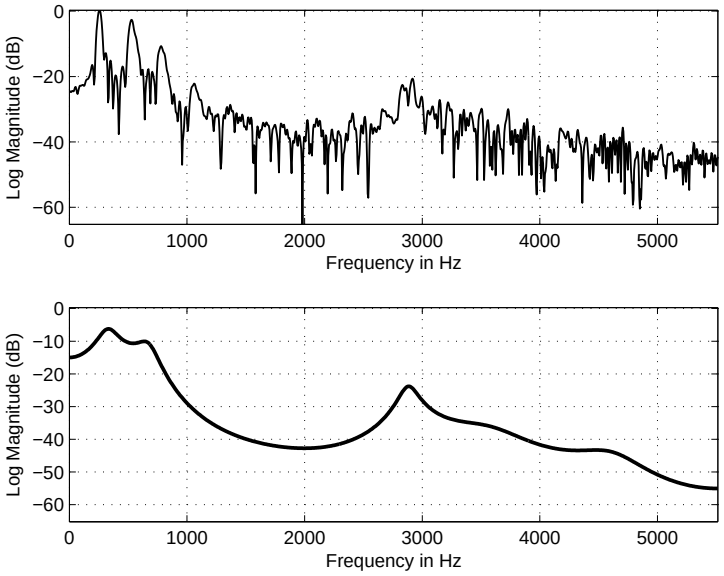

FIGURE 4.4: FFT and spectral estimate from LPC

estimates that can be extracted from the single frame pictures in Figure 4.3. The

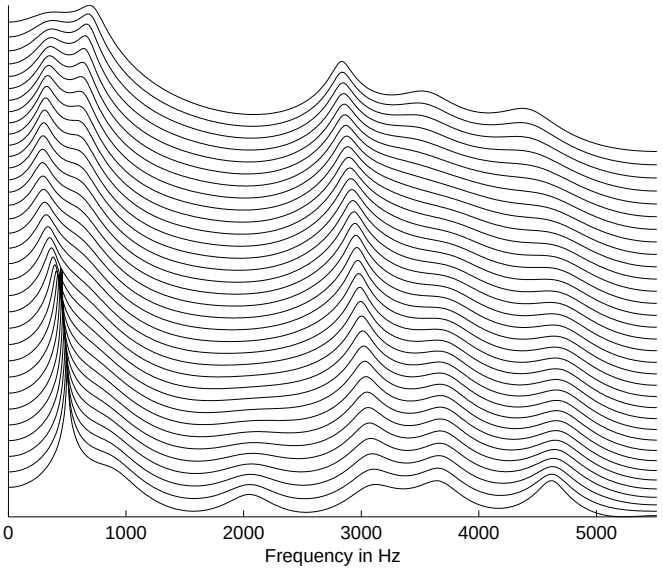

FIGURE 4.5: Line evolution spectrum from TV-LPCC coefficients.

time runs from the bottom going up. It can be clearly seen that the spectrum between the beginning up to the onset of the speech signal is very different from the central spectrum that was captured by the static spectral estimate of the invariant LPC. This shows the advantage of the TV-LPCC in dealing with fast transitions in the signal and will be exploited to enhance the performance of the speech recognition system.

4.5.2 Feature Reduction-based Models

One of the consequences of using time-varying features is that the estimation process is done in a per sample basis rather than per frame. It is then expected that the number of features will not be comparable to the number of features resulting from a nonstationary extraction method. For example, the high resolution provided by the TV-LPCC reflects the huge increase in vectors as shown in Figure 4.6. This increase in the number of feature vectors also poses some numerical calculation problems when dealing with utterance-based averaging. The HMM training procedure could also fail for reasons relating to bursty outliers and numerical computations involving probabilities. Therefore, an effective utilization of the features based on the analysis of the high resolution spectral estimates is required. Figure 4.7 shows an illustration of these two schemes used in this thesis. Other studies

have done feature vector decimation either in the frame length specification or by defining a skipping length during the process to match the number of baseline feature vectors, this is considered as the conventional reduction scheme[47, 48]. In this work spectral difference is used to reduce the number of feature vectors by means of averaging correlated or nearby vectors. This is defined as:

$$SPDIFF = \left(\frac{10}{\ln 10}\right) 2 \left[\sum_{i=1}^{P} \left(c_k^{(i)} - c_k'^{(i)}\right)^2\right]^{1/2} \tag{4.38}$$

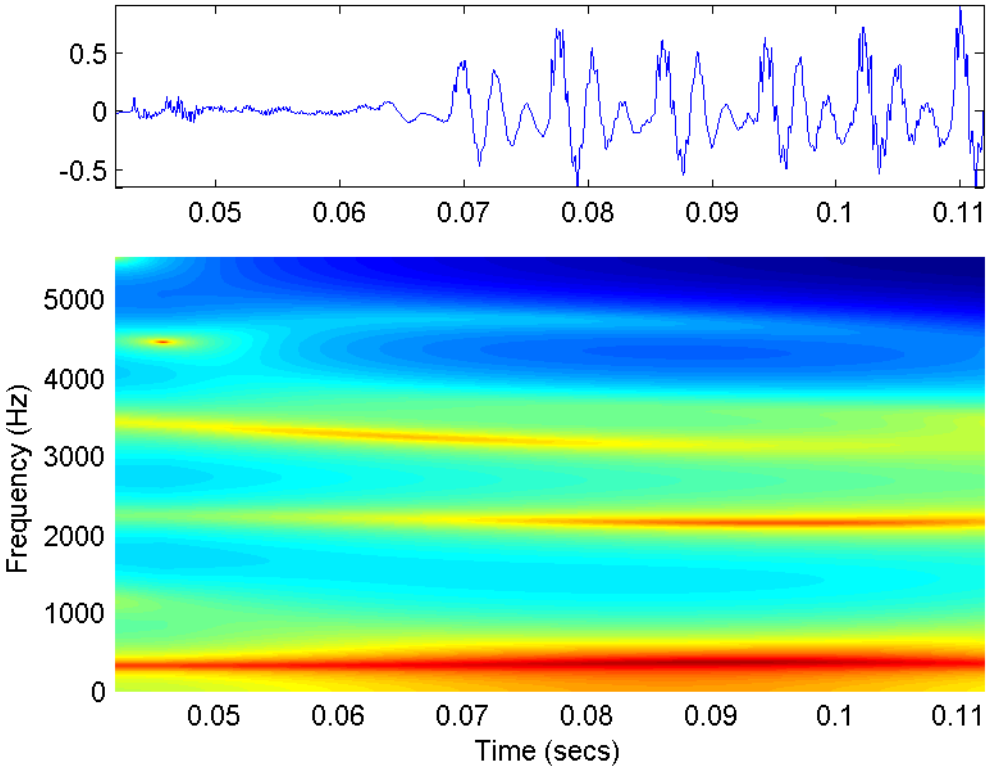

FIGURE 4.6: High resolution spectral estimate from TV-LPCC.

Figure 4.8 illustrates how the number of TV-LPCC vectors are reduced from 512 vectors to 11 vectors via averaging.

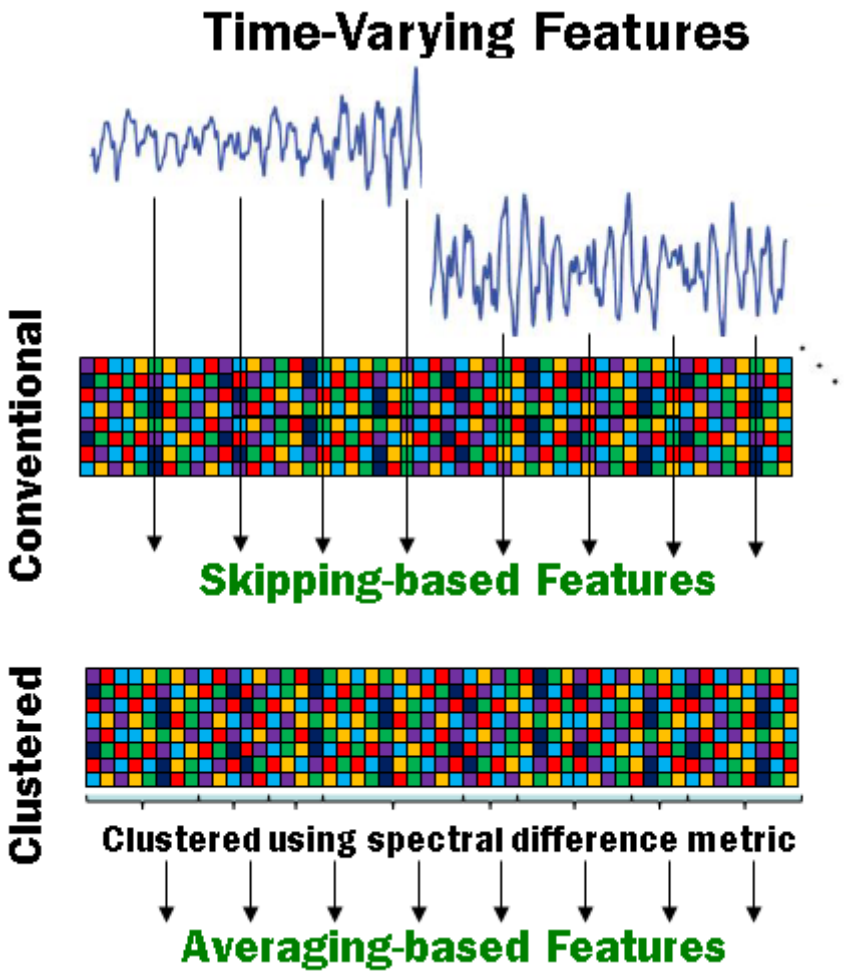

FIGURE 4.7: Feature reduction schemes

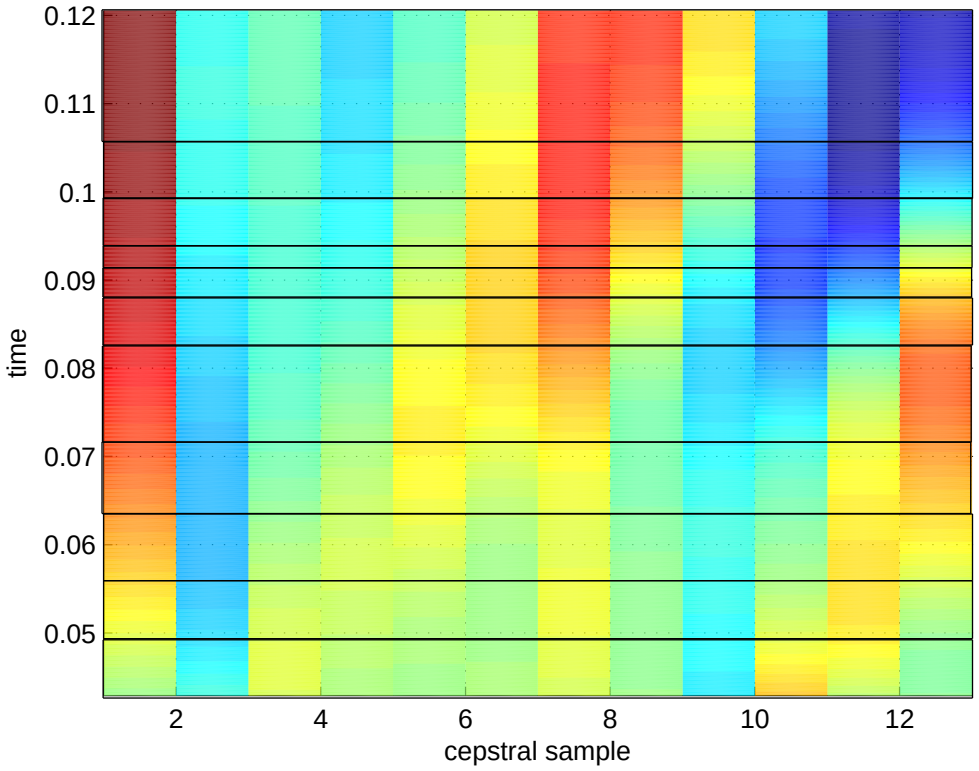

FIGURE 4.8: TV-LPCC clusters based on the SPDIFF as a correlation metric.

Chapter 5

Experimental Setup

5.1 Chapter Overview

This chapter summarizes all the considerations made when conducting the evaluation experiments. Speech recognition is a pattern recognition problem, and as such requires unbiased procedures for assessing the performance of the models involved. For testing models, a 1-fold or hold-out cross-validation scheme is used at an 80-20 training-testing ratio. Variations in the models were made in terms of gender, target words, and training models. Metrics used for assessment are based on general ASR system evaluations such as word accuracy. The database used and the overall parameter settings are also provided.

Noise robustness experiments were also conducted by generating noisy hold-out cross-validation data. The noise generation method was based on a frequency-weighing scheme that avoids low-frequency biasing. Finally, the different systems for noise compensation were enumerated.

5.2 Experimental Measures

5.2.1 Model Selection

Hold-out cross validation will be used to measure the relative performance of the proposed algorithm and its variations. The database D is split into a training

set D_{train} consisting of 80% of the data. The remaining 20% is the hold-out cross validation set $D_{\mathbf{cv}}$. Both training and testing sets are also equal in distribution in terms of gender, each having 32 and 8 speakers, respectively. For each case, the speakers that are included in the training and testing sets are randomly generated and are used consistently for all experiments. The speakers used for the cross-validation set is provided in Table 5.1.

TABLE 5.1: Speakers used for cross-validation

Gender	Speaker Numbers
Male	2, 6, 15, 19, 22, 23, 32, 40
Female	1, 5, 13, 18, 21, 27, 31, 35

5.2.2 Training and Testing Scheme

The training and testing sets described were applied to three separate cases using the baseline formulation: gender-based separation, word-based separation, and model-based. Gender-based separation simply evaluates performance on separate gender sets, word-based separation evaluates performance on words that are causing errors in the baseline system, and model-based pertains to changes in the training model, such as the number of training iterations, changes in the HMM topology, etc.

5.2.3 Evaluation Metrics

For isolated-word speech recognition, the performance accuracy is simply measured using the word error metric. The word error metric is simply the number of substitutions made to the reference words from the test set. While this metric can be considered as too simple for any conclusive hypothesis testing, this is standard performance measurement and is used all through out for comparative conclusions.

5.3 General Setup

All the modules were ran using the MATLAB environment. The parameter settings were based on standard experiments done using the same set of modules for training and recognition.

5.3.1 Database

The training and testing speech data comes from a Japanese language isolated-word command database consisting of 142 words spoken by 40 male and 40 female Japanese speakers. The database spans a subset of words that can be recognized by a human-machine interface robot called Chapit created by the company Raytron. Each word in the database are uttered 4 times by each speaker, although there are a few incidental exceptions. Each files is at exactly 4 seconds of duration with words ranging from 1 to 8 syllables. The files were recorded using a single channel with a sampling frequency of 11025 Hz. The vocabulary list is included in Appendix A for reference. The same database setup and settings described in this chapter was used to evaluate the baseline speech recognition system outlined in Chapter 2.

5.3.2 Parameter Settings

The parameters used for the baseline and proposed models are summarized in Table 5.2. The same back-end recognition system was used. The frame length used for the time-varying speech features was chosen to span approximately the same range as the MFCC. The estimation of the time-varying models was done with $P = 14$ and $Q = 1$, in order to show the minimal performance based on favorable results from other experiments [49]. The basis function used for the TV-LPCC was a simple power function:

$$f_q[n] = n^q \tag{5.1}$$

TABLE 5.2: System summary

	Baseline	**Proposed**
Features	12 MFCC + Log Energy	13 TV-LPCC
Derivatives	$\Delta + \Delta\Delta$	$\Delta + \Delta\Delta$
Frame length	23.2 msec	46.44 msec
Overlap length	11.6 msec	None
Pre-emphasis	Yes	Yes
Window	Hanning	None
Mel-filters	40	N/A
Normalization	Mean, variance	Mean, variance
HMM States	32	32
Viterbi Iterations	10	10

5.4 Noise Robustness Experiments

For the experiments dealing with noise, the inherent robustness of the proposed techniques were first initially tested by not using any noise compensation techniques. Some considerations were also made to the noise model generation in order to avoid spectral biasing.

5.4.1 Noisy Model Generation

Noisy conditions were applied to robustness experiments through the use of the NOISEX noise database. Table 5.3 shows a list of the noise types and the corresponding symbol that will be used for the experimental results.

TABLE 5.3: NOISEX noise types

	Noise Type		Noise Type		Noise Type
N1	Babble	N6	F-16 Cockpit	N11	M109 Tank
N2	Buccaneer Jet Cockpit 1	N7	Factory Floor 1	N12	Machine Gun
N3	Buccaneer Jet Cockpit 2	N8	Factory Floor 2	N13	Pink
N4	Destroyer Engine Room	N9	HF Channel	N14	Volvo Car Interior
N5	Destroyer Operations Room	N10	Leopard Military Vehicle	N15	White

All noise types were mixed with the clean speech samples with SNR levels of 10, 15, and 20 dB. To avoid low-frequency bias, G.712 standard-based filters were applied to the speech before mixing. The number of noise types applied in some cases were reduced due to time constraints. The selection of noise types were based on the relative effect of noisy conditions when frequency weighting is not used. Using

the baseline system, recognition rates were calculated with and without frequency weighting and the differences in accuracies were calculated. The results for this comparison is given in Table 5.4.

TABLE 5.4: Reduction in accuracy rates for frequency-weighted mixing of noise (negative values indicate improvements)

Noise	20 dB	15 dB	10 dB	Noise	20 dB	15 dB	10 dB
N1	5	12	19.2	N8	-2.7	-5.2	-2.5
N2	-0.4	2.9	2.9	N9	6.8	12.9	20.3
N3	-0.7	-2.5	0	N10	9.6	19.6	31.8
N4	-1.3	-0.2	2.6	N12	8.8	11.5	12.5
N5	8.8	15.6	20.6	N13	3.7	7.5	9.4
N6	3	7.2	12.3	N14	7.7	16.5	35.7
N7	6.4	11.9	13.5	N15	-3.5	-5.2	-4.5
N8	7.6	14.5	25.9				

From the table, three categories of noise types were defined based on the relative bias frequency weighting can reduce. Low-bias noise types are within the 5% range: Buccaneer fighter jets noise types, engine room noise, HF radio channel, and white noise. Medium-bias noise types are those in the 5-15% range: F16 fighter jet, factory 1, machine gun, and pink noise. High-bias noise have above 15% reduction influence in recognition rate: voice babble, operations room, factory noise 2, leopard tank, M109 tank, and the Volvo 340 noise types.

5.4.2 Noise Compensation

Finally, the performance of the proposed feature extraction method based on the original formulation are also subject to the noise compensation techniques introduced in Chapter 2. The systems evaluated were CMS/DRA, RASTA/DRA, and RSF/DRA.

Chapter 6

Results and Analysis

6.1 Results and Discussion

6.1.1 Recognition Results

The following tables show the results for the clean cases:

Table 6.1: Average results for clean experiments

Method	Male	Female	Average
MFCC	93.00	92.52	92.76
Skipping	92.30	89.17	90.74
Averaging	92.52	90.34	91.43

It is clearly seen that the results of the experiment show that the proposed schemes are suboptimal when compared to the MFCC.

The following tables are the results for the noisy experiments:

Table 6.2: MFCC baseline (male)

	Baseline			CMS/DRA			RASTA/DRA			RSF/DRA		
Noise Type	10	15	20	10	15	20	10	15	20	10	15	20
N1	44.38	60.68	76.67	50.75	69.71	82.33	48.45	64.54	80.18	51.74	67.89	82.35
N2	61.13	69.63	77.55	72.73	84.04	88.34	66.55	75.68	83.14	71.28	79.62	87.10
N3	67.70	75.37	81.78	71.50	82.98	85.04	70.01	79.86	83.99	71.59	78.97	84.77
N4	73.90	79.07	83.02	81.54	86.56	90.14	77.58	84.68	86.90	80.83	86.14	89.48
N5	44.15	63.99	82.25	45.47	66.73	82.66	44.48	63.60	81.79	44.54	62.84	80.70
N6	72.64	79.73	86.02	77.24	87.10	89.92	74.79	82.10	87.46	76.74	83.16	88.35
N7	57.27	76.79	95.97	57.46	80.12	95.92	57.20	76.06	94.40	55.91	74.06	91.34
N8	74.43	82.31	88.39	75.84	83.43	88.84	74.78	81.92	87.83	74.69	80.34	85.44
N9	72.36	78.76	83.37	75.79	82.63	86.51	73.10	81.64	85.25	73.57	79.70	85.34
N10	77.55	82.17	84.84	82.09	86.68	88.42	79.63	84.35	85.47	81.20	84.07	85.92
N11	67.63	77.01	85.69	72.14	83.36	89.72	69.60	80.53	87.69	70.43	79.82	87.79
N12	59.19	66.89	73.06	66.30	77.18	79.80	60.98	70.96	75.94	62.48	70.40	78.19
N13	68.22	76.87	85.49	71.04	83.42	87.41	70.08	78.76	86.28	70.47	78.98	86.64
N14	77.61	78.70	79.74	79.70	81.12	81.35	78.39	81.78	80.55	77.28	79.34	80.73
N15	69.59	77.24	84.52	70.66	78.91	84.99	69.75	78.03	84.45	69.01	77.01	83.93
Average	65.85	75.01	83.22	70.02	80.93	86.76	67.69	77.63	84.75	68.78	77.49	85.20

Table 6.3: MFCC baseline (female)

	Baseline			CMS/DRA			RASTA/DRA			RSF/DRA		
Noise Type	10	15	20	10	15	20	10	15	20	10	15	20
N1	42.95	60.23	75.53	47.76	65.69	81.34	46.88	64.01	79.33	45.17	61.76	77.23
N2	62.49	71.81	79.88	66.75	80.26	87.10	65.58	75.53	85.20	63.80	74.03	83.23
N3	55.71	62.96	69.98	59.53	72.07	78.57	58.32	67.76	76.92	56.54	66.37	74.68
N4	76.34	81.27	86.02	80.37	89.14	90.29	80.62	85.27	88.68	79.69	84.15	87.07
N5	40.77	60.08	78.97	45.52	66.59	85.01	44.64	64.54	83.44	43.53	62.71	81.73
N6	70.27	78.03	84.10	75.13	84.94	86.84	75.06	81.65	86.57	73.73	79.88	84.84
N7	47.39	67.39	86.27	52.12	73.15	84.57	51.29	68.71	85.29	49.71	67.30	84.54
N8	71.88	79.33	86.55	76.81	86.92	89.51	75.13	82.77	89.33	72.68	81.42	88.59
N9	70.71	76.65	82.31	73.81	80.94	88.03	73.43	81.32	88.27	71.94	79.94	86.57
N10	75.80	80.46	83.94	80.71	87.70	87.83	79.71	83.44	86.62	78.63	82.60	85.15
N11	66.38	75.73	84.68	70.57	82.32	87.78	70.37	79.52	87.09	69.21	78.10	86.31
N12	58.12	66.12	72.50	61.44	72.11	74.67	60.34	67.88	74.99	58.49	66.50	73.76
N13	57.78	66.74	75.22	61.97	77.89	83.87	60.75	72.34	82.56	58.87	69.90	79.67
N14	76.07	77.73	78.24	80.12	81.66	82.24	80.01	81.11	81.80	79.25	81.01	81.18
N15	58.36	66.67	73.99	63.34	75.32	81.49	62.46	72.04	80.40	61.40	69.64	77.78
Average	62.07	71.41	79.88	66.40	78.45	84.61	65.64	75.19	83.77	64.18	73.69	82.16

Table 6.4: TV-LPCC skipping (male)

	Baseline			CMS/DRA			RASTA/DRA			RSF/DRA		
Noise Type	10	15	20	10	15	20	10	15	20	10	15	20
N1	37.85	55.41	71.13	46.88	63.00	77.46	45.17	61.49	76.52	43.13	59.24	74.71
N2	52.00	59.45	65.53	61.75	73.15	83.87	59.67	70.52	80.29	57.18	67.01	75.00
N3	60.15	67.40	74.03	64.04	73.25	82.13	63.74	72.71	80.63	62.45	71.17	78.20
N4	67.52	72.21	75.87	75.42	82.28	87.15	73.77	80.44	85.40	71.82	77.33	82.28
N5	35.77	56.34	75.62	38.46	57.84	77.05	38.52	58.63	77.09	37.74	58.30	77.04
N6	66.77	72.36	76.62	75.98	80.73	84.37	74.38	79.62	83.55	71.47	76.96	81.27
N7	52.61	70.51	86.92	56.24	74.87	91.74	55.98	73.88	91.07	54.50	73.07	90.31
N8	67.16	74.19	80.10	74.99	80.31	83.99	74.04	79.75	83.55	72.04	77.43	82.17
N9	68.27	72.97	75.96	70.07	77.68	83.41	70.21	76.54	82.67	69.82	75.37	80.21
N10	70.65	73.90	76.61	79.90	82.24	84.34	78.57	81.07	83.14	75.68	79.31	81.20
N11	62.14	70.43	77.75	70.70	77.95	84.65	68.79	76.82	83.47	66.84	75.03	81.47
N12	51.68	59.13	65.23	65.60	70.93	75.69	63.31	69.29	73.66	59.46	65.72	70.83
N13	63.97	72.93	81.20	65.82	76.51	85.56	66.39	76.30	85.40	65.47	75.25	83.58
N14	72.04	73.46	72.90	76.88	76.94	75.59	75.85	76.71	75.93	74.73	75.18	74.62
N15	57.75	67.41	75.62	60.98	71.31	81.05	61.32	70.96	80.24	59.93	69.33	78.58
Average	59.09	67.87	75.41	65.58	74.60	82.54	64.65	73.65	81.51	62.82	71.71	79.43

Table 6.5: TV-LPCC skipping (female)

	Baseline			CMS/DRA			RASTA/DRA			RSF/DRA		
Noise Type	10	15	20	10	15	20	10	15	20	10	15	20
N1	35.24	53.60	70.73	44.15	61.19	77.51	42.65	60.26	76.19	39.74	57.10	74.32
N2	55.78	65.44	74.35	55.11	68.70	81.93	55.71	69.00	81.03	56.24	68.31	78.81
N3	50.96	57.43	63.67	53.24	62.80	72.14	53.42	62.95	71.14	52.20	60.88	68.51
N4	69.20	74.66	78.84	70.55	78.35	85.98	70.92	78.38	84.84	70.80	77.36	82.47
N5	32.60	54.68	75.55	39.66	61.03	81.32	39.00	60.38	80.26	36.98	58.10	79.15
N6	63.75	70.41	76.21	68.59	75.91	82.23	68.46	75.57	81.32	66.45	73.07	79.27
N7	41.58	61.47	81.16	45.89	63.06	78.81	45.64	64.00	80.42	44.65	63.30	80.67
N8	65.41	74.28	81.38	71.88	77.85	83.06	71.44	78.52	83.93	69.51	77.12	83.07
N9	63.25	70.97	77.88	63.84	73.53	82.24	64.24	73.95	82.38	63.94	73.24	80.59
N10	70.34	74.61	77.37	74.31	78.77	82.83	74.13	78.56	82.48	73.30	76.82	80.23
N11	58.77	69.19	78.94	66.06	75.19	82.83	64.61	74.09	82.80	62.63	72.37	81.55
N12	53.07	60.13	66.93	54.67	63.35	70.69	54.35	63.13	70.42	53.92	62.15	69.30
N13	51.18	61.43	70.17	58.36	69.76	80.17	57.57	68.36	78.59	55.41	66.46	75.95
N14	68.97	70.63	71.26	75.76	76.81	77.49	75.02	75.93	76.80	72.99	75.05	75.25
N15	51.69	60.56	67.63	55.83	65.37	74.69	55.74	64.62	73.41	54.73	63.95	71.67
Average	55.45	65.30	74.14	59.86	70.11	79.59	59.53	69.85	79.07	58.23	68.35	77.39

Table 6.6: TV-LPCC averaging (male)

	Baseline			CMS/DRA			RASTA/DRA			RSF/DRA		
Noise Type	10	15	20	10	15	20	10	15	20	10	15	20
N1	37.51	55.17	70.93	45.64	61.88	77.24	44.31	60.59	76.52	41.89	58.75	75.03
N2	53.20	60.62	66.18	62.37	73.52	84.55	61.44	72.02	81.01	58.60	67.70	76.34
N3	61.87	69.21	74.75	66.26	75.11	82.90	66.27	74.62	81.54	64.99	72.40	79.06
N4	69.11	73.52	76.29	76.80	82.30	87.68	75.35	81.04	85.42	73.86	78.41	82.06
N5	37.22	57.45	75.99	40.45	59.31	77.46	40.56	59.45	77.45	39.09	59.14	77.20
N6	68.06	73.14	76.98	77.77	82.61	85.54	75.59	80.41	83.88	73.22	78.34	81.80
N7	54.10	71.23	87.32	57.46	75.78	92.41	57.26	74.94	92.00	56.62	74.26	90.09
N8	68.44	74.94	80.36	76.84	81.14	85.10	75.29	80.70	84.86	72.90	77.96	82.79
N9	69.72	73.62	76.36	71.54	78.01	83.82	72.37	77.78	82.51	71.56	76.94	80.59
N10	71.59	74.50	77.00	81.43	84.27	85.18	80.00	82.10	83.42	77.37	79.47	81.39
N11	63.41	71.68	78.03	71.84	79.37	85.41	71.14	78.18	83.97	68.56	76.10	82.07
N12	52.72	59.90	65.64	66.94	72.52	76.62	63.91	69.81	74.35	60.29	67.01	71.75
N13	67.22	74.34	81.34	68.35	77.39	85.70	68.35	77.13	85.52	68.21	76.12	83.74
N14	73.25	73.93	73.19	78.77	78.34	76.57	77.78	77.47	76.86	76.65	76.22	75.70
N15	59.19	68.03	75.77	62.48	72.53	81.21	62.73	72.54	80.80	61.13	70.41	79.07
Average	60.44	68.75	75.74	67.00	75.60	83.16	66.16	74.59	82.01	64.33	72.61	79.91

Table 6.7: TV-LPCC averaging (female)

	Baseline			CMS/DRA			RASTA/DRA			RSF/DRA		
Noise Type	10	15	20	10	15	20	10	15	20	10	15	20
N1	35.63	53.94	71.48	44.65	61.54	78.15	43.38	60.42	77.13	40.74	58.32	75.72
N2	57.17	66.32	75.00	56.54	70.29	82.57	57.51	69.47	81.36	57.33	68.14	78.79
N3	53.28	59.71	65.43	55.39	65.20	74.03	55.73	64.81	72.83	55.19	62.88	70.05
N4	71.91	76.32	79.69	72.73	80.22	87.21	73.71	80.07	86.25	73.03	78.47	83.67
N5	33.83	55.62	76.73	41.55	62.30	82.60	40.48	62.20	82.19	37.99	59.21	79.83
N6	65.54	71.83	77.49	69.99	77.41	83.93	69.60	77.20	83.08	68.06	74.66	80.92
N7	43.55	63.45	82.41	47.42	64.04	80.29	47.81	64.59	81.24	46.23	64.21	81.79
N8	66.59	74.80	82.56	73.69	79.69	84.74	72.63	79.50	84.84	70.74	77.75	83.82
N9	66.20	73.46	79.21	66.40	75.38	83.52	67.13	75.47	83.02	66.77	74.72	81.46
N10	72.58	76.07	78.92	76.43	81.14	84.24	76.75	80.56	83.82	75.30	79.06	81.68
N11	60.61	70.71	80.04	67.48	76.01	83.93	66.24	75.37	84.00	64.09	73.46	82.09
N12	54.98	62.38	68.27	56.77	65.21	72.35	57.43	65.44	71.92	56.31	64.25	70.74
N13	53.71	63.07	71.09	60.50	71.18	81.03	58.99	69.61	79.83	57.26	67.59	76.64
N14	70.76	72.26	72.26	77.21	78.49	78.89	76.77	77.45	77.84	74.79	76.23	76.23
N15	53.51	61.63	69.44	57.94	67.30	76.65	57.66	67.45	75.43	55.90	65.05	73.45
Average	57.32	66.77	75.33	61.65	71.69	80.94	61.45	71.31	80.32	59.98	69.60	78.46

6.1.2 Discussion of General Findings

From the experiments, it is the sensitivity of the TV-LPCC that is contributing to its degraded performance. A small difference in the samples were observed to lead to big differences in the resulting cepstra. To show this, we compared the trajectories of the time-varying coefficients for the case when boundaries were generated automatically and when it is manually corrected to match the boundaries used for training the same speech. It is clearly seen in Figure 6.1 that the shift caused by the blocking procedure could result to the trajectories of the same noisy signal to become different.

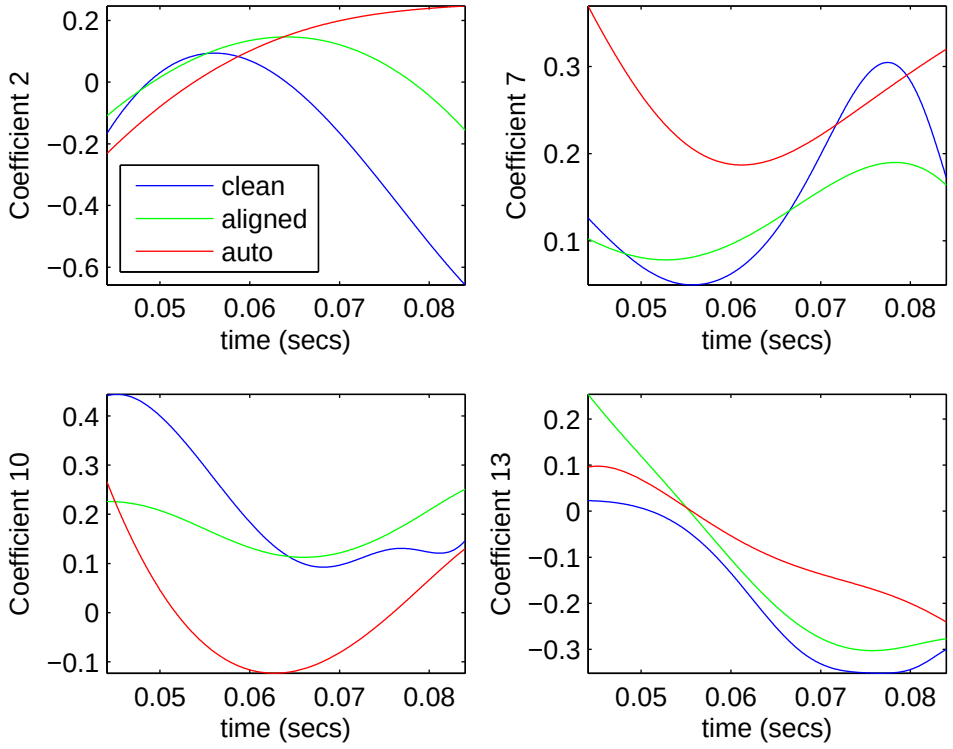

FIGURE 6.1: Effect of VAD performance to cepstrum trajectories.

Thus, for all the experiments, the result of the VAD for clean speech signals had to be used for all the noisy test cases. However, as shown in Tables 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, and 6.7, both proposed techniques always performed worse than the baseline. For all cases, CMS with DRA has the best average performance and is used for succeeding experiments.

One interesting observation is the comparison of performance between the skipping and the averaging schemes. In general, the averaging reduction method performs better than skipping making the proposed method one step ahead of the conventional way of reducing the features. Despite the suboptimal performance, the fact that the features can recognize words makes the results encouraging for future work.

6.2 Hybrid Models

Based on the results, it can also be argued that the standard MFCC formulation is already performing well for the case of voiced speech segments and the use of nonstationary modeling can be used as an augmentation for prevalent problems. It is common practice in the speech recognition literature to combine certain models using a justified scheme. For this research, two possibilities in combining stationary and nonstationary features were employed. The first is collectively called data selective, which is based on the notion that transient parts of the signal are required to have better modeling. The other group is based on the notion that different features can have varying strengths in dealing with different speech phenomena as employed in several works in experimental speech recognition.

6.2.1 Voting-Based Models

The idea of voting-based methods comes from the fact that different features may have varying properties that will be effective for different set of speech phenomenon. This is especially true for noisy cases where modulation and varying spectral influences are present. Because a number of models is generated from the baseline formulation of the time-varying methods, consensus of different systems can be gathered where the most frequent recognition can be used as the overall system output. This type of scheme is depicted in Figure 6.2.

Using the training and testing set, the performance of MFCC and skipping TV-LPCC were tested on the group of words *genki*, *tenki*, and *denki*. This comparison was done by taking the entire duration of speech with a few silence included by the energy-based VAD. The original 32-state left-to-right HMM was used.

For the second case, the speech signal was cut in half. Due to the inconsistencies with respect to the cut-off of the speech signal. The HMM model was modified such that it can reflect this inconsistency. First, a few speech files were analyzed and the coverage of the signal were found to vary between getting only the first consonant up to getting until the /n/ or /k/ sound. Based on this, 3-state were used to span one phoneme. Including the initial silence, 16 states will be used. Every third state is also let skip to the end in case the speech is cut off immediately. This is shown in Figure 6.3.

For both cases, the use of the average TV-LPCC derived cepstral coefficients prove to have better distinction between the P, T, and G sounds after enough training iterations as shown in Figures 6.4 and 6.5. The margins are very large. To confirm the hypothesis, a small experiment was done using the same set-up as outlined in Chapter 4. However, it uses a parallel recognition scheme, where the consensus is used as the actual output. This is shown in Figure 6.6. TVLPCC1 refers to TV-LPCC using Skipping and TVLPCC2 referes to TV-LPCC using Averaging.

The results of the experiment is summarized in Table 6.8. Note that only the Babble, Buccaneer 1 and 2, Destroyer engine room, HF channel, Pink, and White Noise types were included due to time constraints. Also, this result is only for male speakers. As can be seen, an average of 1.02% increase in recognition rate was achieved. This confirms the observation that the sensitivity of the features is also affected by speaker-dependence and other factors.

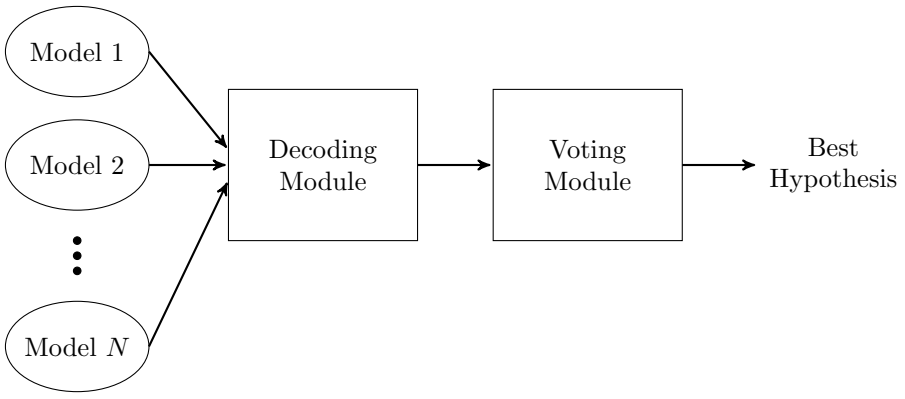

FIGURE 6.2: Voting-based scheme architecture

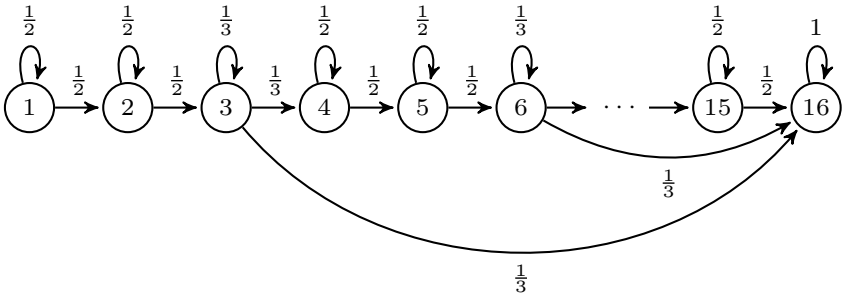

FIGURE 6.3: The HMM topology modified for cut onsets.

6.2.2 Data Selective Models

Based on the analysis done in evaluating the performance of the best system for the HU-SCS, words causing the most errors only differ by one phoneme specifically at the beginning or at the onset. The idea for these data selective models

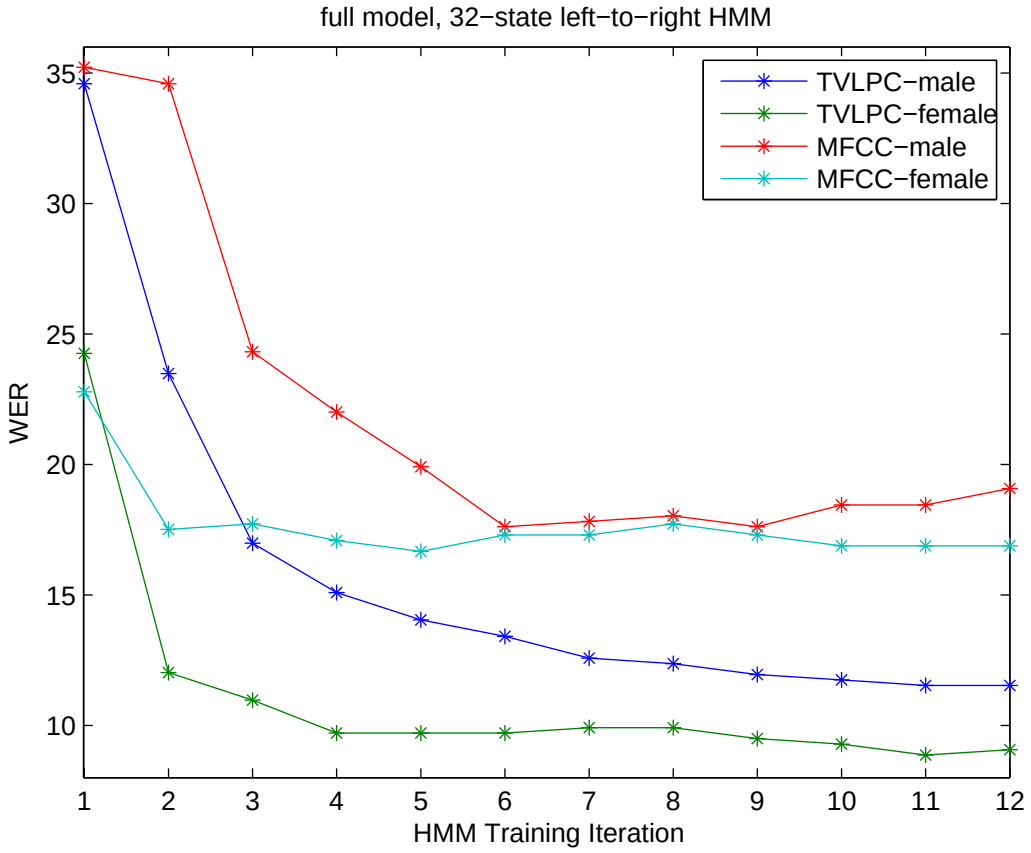

FIGURE 6.4: MFCC and TV-LPCC performance on closely sounding words using a 32-state HMM.

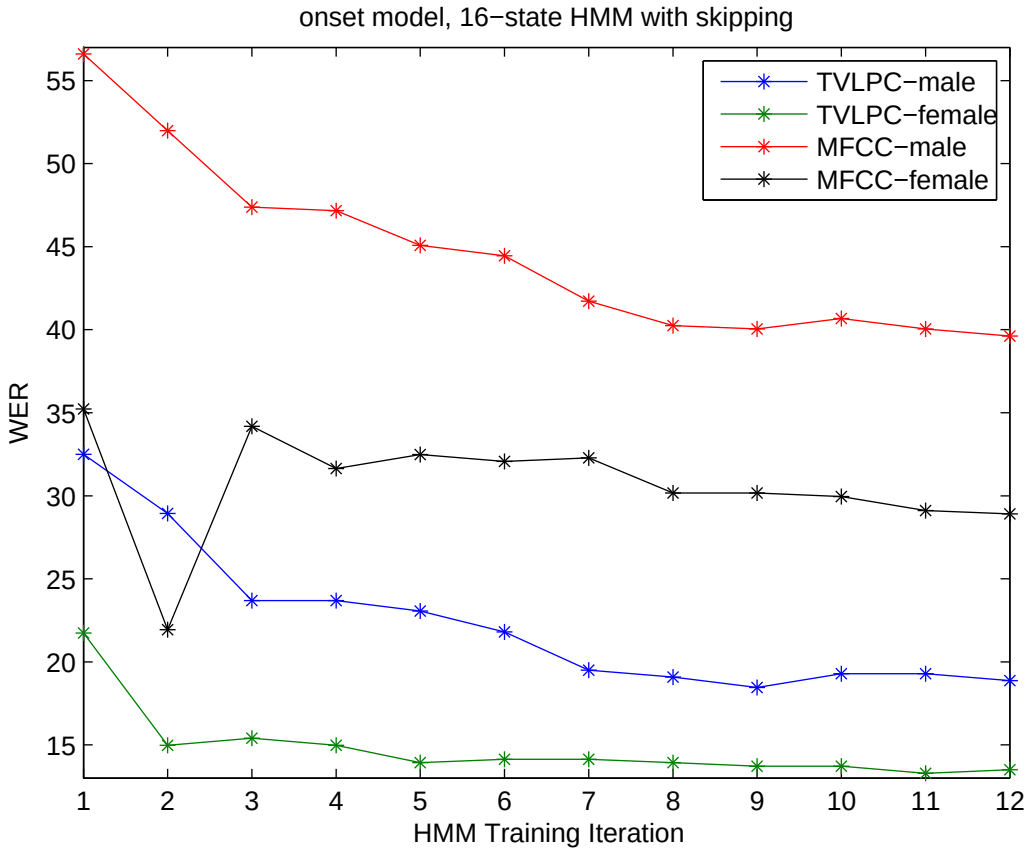

FIGURE 6.5: MFCC and TV-LPCC performance on closely sounding words (halved in time) using a 16-state skipping HMM.

is to incorporate both time-varying and invariant features on the training data. The idea is to only apply time-varying analysis to transient parts of the speech

TABLE 6.8: Average results for voting-based experiment

Noise types	MFCC	MFCC+TVLPCC1+TVLPCC2
Clean	93.00	95.82
N1	82.33	83.74
N2	88.34	88.34
N3	85.04	85.32
N4	90.14	92.39
N9	86.51	87.36
N13	87.41	87.69
N15	84.99	85.27
Avg	87.22	88.24

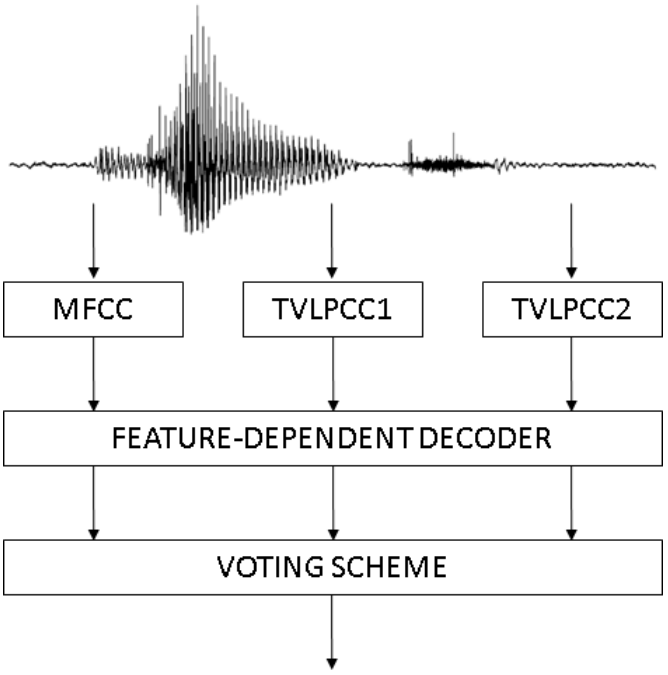

FIGURE 6.6: Voting-based system used

signal and apply the original stationary formulation to the rest of the data. However, considering the sensitivity of the time-varying method, a precise detection of end-points and glottal changes must be in place. This is similar to studies that made use of time-varying techniques to classify CV transitions, where this type of selectivity has been proven to be effective [50].

For this research a VAD-dependent scheme is used. Instead of using an end-point detection algorithm, a minimal constant number of frames is determined via an iteration process. This iteration process is only applied to set of utterances that are considered to require the use of the time-varying models, such as the close sounding word problem being solved. The procedure computes for the time-varying features using the best performing baseline model among the non-hybrid types using an increasing number of frames for every set, and the smallest number of frames where all sets are performing well is used. Once the constant number of frames is determined, the division will be applied to all words in the database for evaluation. Figure 6.7 shows the result of this procedure to two near sounding groups.

To continue this process, we have settled for 7 frames to confirm the hypothesis. Shown in Figure 6.8 is the scheme used for the experiment. Initially, 7 frames

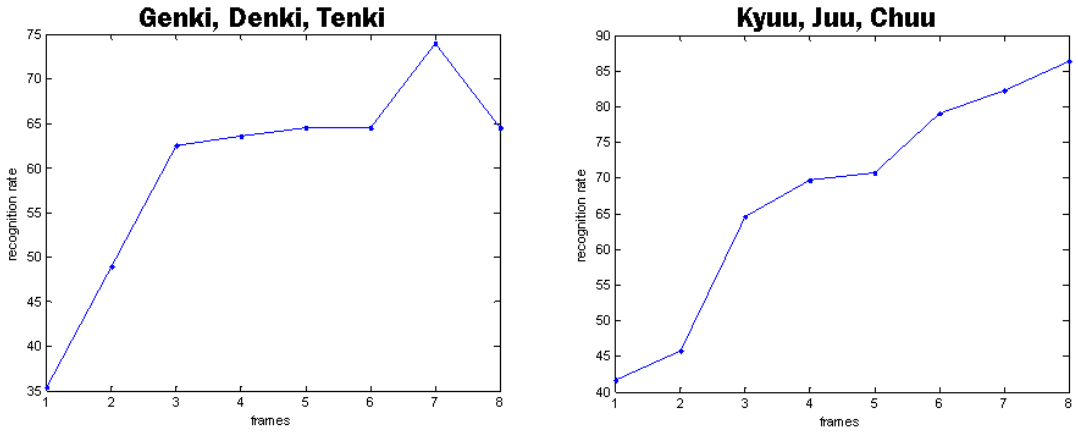

FIGURE 6.7: Near sounding word recognition as a function of number of frames used.

are used to solve for the TV-LPCC features. Then, the rest of the signal was subject to MFCC extraction. Similar to the averaging technique, using Equation 4.38, the number of TV-LPCC vectors were reduced in order to make training feasible. The results were then compared to the new hybrid schemes proposed. Clearly, both hybrid schemes improve the performance of the baseline system by 1-2%. However, it should be noted that this assumes that the system knows the appropriate boundaries as detected for clean speech.

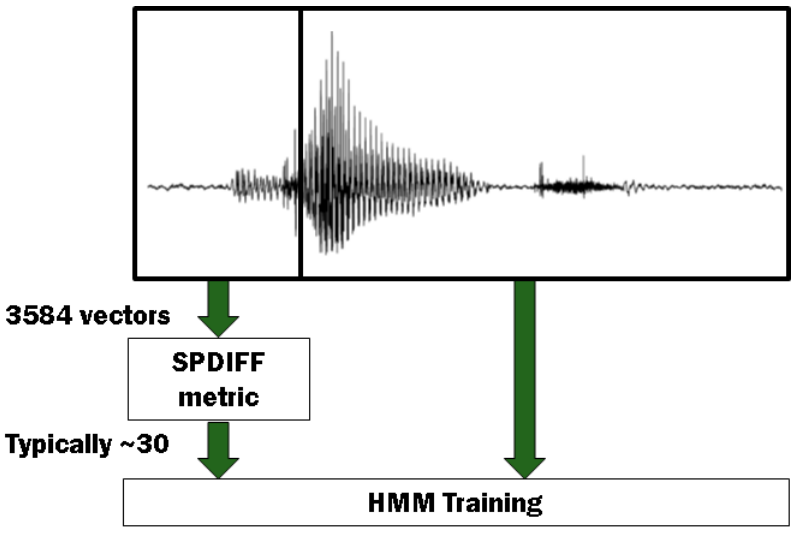

FIGURE 6.8: Split signal data-selective model scheme.

Table 6.9: Average results for hybrid experiments

		White		Babble		
Noise types	Clean	10	20	10	20	Average
MFCC	93	70.66	84.99	50.75	82.33	76.35
Voting-Based	95.82	71.21	85.27	52.04	83.74	77.62
Split Features	94.02	72.37	87.07	53.46	86.11	78.61

Chapter 7

Conclusion and Future Direction

7.1 Summary of Thesis

The research done in this thesis focused on the use of time-varying feature extraction for speech recognition. The research background and motivation pertaining to speech modeling and speech recognition were presented, and the theoretical and practical issues involved in the development process. Highlighting modeling accuracy and low complexity, the basic formulation of time-varying LPC coefficients was described and successfully implemented. The initial experiments have proven to be suboptimal due to sensitivity issues but further analysis based on further experiments have led to hybrid models that give marginal gains by exploiting both the strength and weaknesses of the proposed technique. Overall, the result of the experiments have shown that there is indeed some potential in the use of TV-LPCC as speech features.

7.1.1 Contributions

This work presented a pioneering work where the use of an HMM-based, isolated-word speech recognition system with words as acoustic units is exploited to discover and highlight the merits of time-varying LPCC as features. In particular, the gains acquired by incorporating the technique did not require very high complexity since the proposed technique can be solved using linear operations.

The thesis was successful in its implementation of the time-varying speech feature extraction method as the system can recognize words using the proposed scheme. However, using the proposed formulation did not surpass the performance of the currently best-performing system. This is only true if the proposed scheme is used for the entirety of the speech signal.

The thesis also provided several experimental results for noisy test conditions. The interaction of the proposed scheme with various feature compensation techniques were tabulated and recorded. It was concluded that the reason behind the suboptimality is the sensitivity of the feature extraction method.

Finally, these aforementioned sensitivity and instability issues associated with the technique have been worked around by data-selective splitting of features, reducing the occurrences of bursts in cepstral values that increase the model entropy. Small-scale experiments were conducted in order to verify the hypothesis. It is then concluded that the time-varying feature extraction scheme can be well-suited for short-duration signals. Hybrid systems were proposed, combining the merits of both time-varying and time invariant feature extraction scheme and the results were positive.

7.1.2 Results Summary

The use of time-varying LPCC features has been successfully implemented for use in an isolated-word speech recognition system. The evaluation experiments have shown that the inherent sensitivity of the technique prevents it from performing better than MFCC due to the bursty spectral estimates. However, despite this, it was found that it can solve the problem of near similar pronunciation words by not using entire words, but only the transient parts of the signal.

A series of simulations were performed using training data and cross-validation data. Noise robustness experiments were performed under clean, 20 dB, 15 dB, and 10 dB SNR conditions. In all cases, the proposed technique did not perform better than the baseline feature technique. By similar argument, the results for TV-LPCC are not representative of the noise robustness of the technique but is attributed to the sensitivity and burstiness. This was proven when the same

procedure was applied using only the near similar pronunciation words as the TV-LPCC performed better than the MFCC in the latter cases.

Finally, by doing separate experiments, gains in the use of time-varying speech features were acquired when the TV-LPCC features were used alongside MFCC. This was motivated by using voting-based system that also improved the performance of the system due to errors from similar pronunciation words.

7.2 Recommendations for Future Work

The most natural step in continuing this work is to pinpoint areas in which parameter settings were being made empirically. Recently, lots of work are being done in the field of Artificial Intelligence where most of these complex optimization are being done automatically. For example, simulated annealing and other genetic algorithms can be used to optimize all the parameters of the system including the LPC order and basis function order.

Another issue that may arise is in the fair comparison between time invariant and time-varying schemes. In this work, the frame lengths for the time-varying scheme was chosen such that the skipping models will output vectors that are the center of the frames for the time invariant cases. This does not necessarily reflect the strength of the time-varying method. Better reduction methods can also be employed such that no information is discarded. This can be thought of as a part of Missing Feature Theory where the most relevant information from the high resolution cepstrum is computed. Both the skipping and averaging methods introduced in this thesis are lossy and there are no justified reason for using them except that one is conventional and the other one is straightforward.

Another area that can be approached is in smoothing the bursty spectral and cepstral estimates of the time-varying LPC. This can drastically change the performance of the technique given that even with this bursty estimate, the accuracy for clean settings is still high for the proposed method. This can further lead to the investigation of the entropy of the proposed scheme. Using an entropy measure, a better way of doing split models can be developed. This idea is fairly new to speech recognition and further study of hybrid schemes will definitely be beneficial.

Appendix A

Vocabulary List

The following are the 142 words contained in the database used for training and testing the systems in this thesis:

1. ohayou
2. konnichiwa
3. konbanwa
4. oyasumi
5. ittekimasu
6. tadaima
7. baibai
8. matane
9. sayounara
10. hajimemashite
11. kawaiine
12. ikutsu
13. onamaewa
14. genki
15. saikindou
16. tsukareta
17. omoronai
18. suki
19. kirai
20. tanoshiine
21. kanashiine
22. tsuraine
23. tsumaranai
24. eraine
25. omoshiroine
26. kashikoine
27. ureshiine
28. nemuine
29. mukatsuku
30. samuine
31. atsuine
32. arigatou
33. utte
34. shoumei
35. terebi
36. bideo
37. eakon
38. sutando
39. dibidi (DVD)
40. denki
41. shoumeitsukete
42. shoumeikeshite

43. terebitsukete
44. terebitsukete
45. bideotsukete
46. bideokeshite
47. eakontsukete
48. eakonkeshite
49. channeru
50. onryou
51. boryuumu
52. myuuto
53. dengen
54. saisei
55. teishi
56. ichijiteishi
57. yoyaku
58. ao
59. aka
60. midori
61. kiiro
62. biesu (BS)
63. shiesu (CS)
64. haadodisuku
65. senkyoku
66. bangumihyou
67. hyouji
68. dejitaruterebi
69. deetahousou
70. housoukirikae
71. nyuuryokukirikae
72. shouon
73. modoru
74. subete
75. menyuu
76. soufuu
77. reibou
78. danbou
79. jidou
80. joshitsu
81. dorai
82. ondo
83. kazamuki
84. taimaa
85. tsukete
86. keshite
87. sutaato
88. sutoppu
89. akaruku
90. kuraku
91. kaishi
92. kakunin
93. henkou
94. settei
95. kaijou
96. kanryou
97. kettei
98. tasukete
99. tsugi
100. chappito
101. tenki
102. zero
103. ichi
104. ni
105. san
106. yon
107. go
108. roku
109. nana
110. hachi
111. kyuu
112. ku
113. juu
114. juuichi
115. juuni

116. hai
117. iie
118. shuuryou
119. torikeshi
120. owari
121. zenshin
122. koutai
123. mae
124. ushiro
125. on (ON)
126. offu (OFF)
127. appu (UP)
128. daun (DOWN)
129. akeru
130. shimeru
131. dai
132. chuu
133. shou
134. ue
135. shita
136. migi
137. hidari
138. kyou
139. jaku
140. koutsuu
141. kankou
142. annai

Bibliography

[1] Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(4):745–777, 2014.

[2] Yifan Gong. Speech recognition in noisy environments: A survey. *Speech communication*, 16(3):261–291, 1995.

[3] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al. Automatic speech recognition and speech variability: A review. *Speech Communication*, 49(10):763–786, 2007.

[4] Yoshikazu Miyanaga, Wataru Takahashi, and Shingo Yoshizawa. A robust speech communication into smart info-media system. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E96-A(11):2074–2080, 2013.

[5] Shingo Yoshizawa, Naoya Wada, Noboru Hayasaka, and Yoshikazu Miyanaga. Scalable architecture for word HMM-based speech recognition and VLSI implementation in complete system. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 53(1):70–77, Jan 2006. ISSN 1549-8328. doi: 10.1109/TCSI.2005.854408.

[6] Korbinian Riedhammer, Van Hai Do, and James Hieronymus. A study on LVCSR and keyword search for Tagalog. In Frédéric Bimbot, Christophe Cerisara, Cécile Fougeron, Guillaume Gravier, Lori Lamel, François Pellegrino, and Pascal Perrier, editors, *INTERSPEECH*, pages 2529–2533. ISCA, 2013. URL http://dblp.uni-trier.de/db/conf/interspeech/interspeech2013.html#RiedhammerDH13.

[7] Rowena Cristina Guevara, Melvin Co, Emerson Tan, Ian Garcia, Evan Espina, Ryan Ensomo, and Ramil Sagum. Development of a Filipino speech corpus. In *Proc. 3rd National ECE Conference*, 2002.

[8] Ricardo Maria Nolasco. Filipino and Tagalog, not so simple. http://www.dalityapi.com/2007/08/articles-filipino-and-tagalog-not-so.html, 2007. Accessed June 2013.

[9] Federic Ang, Juan Carlo Miguel Ancheta, Karmela Mariz Francia, and Krisel Chua. Evaluation of smoothing techniques for language modeling in automatic Filipino speech recognition. In *TENCON 2012 - 2012 IEEE Region 10 Conference*, pages 1–5, 2012. doi: 10.1109/TENCON.2012.6412249.

[10] Federico Ang, Maria Czarina Burgos, and Marvin De Lara. Automatic speech recognition for closed-captioning of Filipino news broadcasts. In *Natural Language Processing and Knowledge Engineering (NLP-KE), 2011 7th International Conference on*, pages 328–333, 2011. doi: 10.1109/NLPKE.2011.6138219.

[11] Ramil Sagum, Ryan Ensomo, Emerson Tan, and Rowena Cristina Guevara. Phoneme alignment of Filipino speech corpus. In *TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region*, volume 3, pages 964–968 Vol.3, 2003. doi: 10.1109/TENCON.2003.1273390.

[12] Sakriani Sakti, Ryosuke Isotani, Hisashi Kawai, and Satoshi Nakamura. The use of Indonesian speech corpora for developing Filipino continuous speech recognition system. In *in Proc. O-COCOSDA*, pages 56–61, November 2010.

[13] Puming Zhan and Martin Westphal. Speaker normalization based on frequency warping. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 2, pages 1039–1042 vol.2, 1997. doi: 10.1109/ICASSP.1997.596118.

[14] Timo Kaukoranta, Pasi Fränti, and Olli Nevalainen. Iterative split-and-merge algorithm for VQ codebook generation. In *Optical Engineering*, volume 37, pages 2726–2732. 1998.

[15] Mark Gales. Semi-tied covariance matrices for hidden Markov models. *Speech and Audio Processing, IEEE Transactions on*, 7(3):272–281, 1999. ISSN 1063-6676. doi: 10.1109/89.759034.

[16] Hua Yu and Alex Waibel. Streamlining the front end of a speech recognizer. In *In ICSLP 2000*, volume 1, pages 353–356, 2000.

[17] Eric Ristad. A natural law of succession. Technical report, Comp. Sci. Dept., Princeton Univ., 1995. CS-TR-495-95.

[18] Slava Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 35(3):400–401, 1987. ISSN 0096-3518. doi: 10.1109/TASSP.1987.1165125.

[19] Ian H. Witten and Timothy Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *Information Theory, IEEE Transactions on*, 37(4):1085–1094, 1991. ISSN 0018-9448. doi: 10.1109/18.87000.

[20] Hermann Ney and Ute Essen. On smoothing techniques for bigram-based natural language modelling. In *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, pages 825–828 vol.2, 1991. doi: 10.1109/ICASSP.1991.150464.

[21] Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 1, pages 181–184 vol.1, 1995. doi: 10.1109/ICASSP.1995.479394.

[22] Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 310–318. Association for Computational Linguistics, 1996.

[23] Mark Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12:75–98, 1998.

[24] Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel. A one-pass decoder based on polymorphic linguistic context assignment. In *in Proc. of ASRU*, 2001.

[25] Joel Ilao and Rowena Cristina Guevara. Investigating spelling variants and conventionalization rates in the Philippine national language's system of

orthography using a Philippine historical text corpus. In *in Proc. of O-COCOSDA*, December 2012.

[26] Oron Gamliel and Ilan Shallom. Perceptual time varying linear prediction model for speech applications. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 4601–4604. IEEE, 2009.

[27] Aki Härmä, Marko Juntunen, and Jari Kaipio. Time-varying autoregressive modeling of audio and speech signals. In *Proceedings of the EUSIPCO*, pages 2037–2040. Citeseer, 2000.

[28] Milan Milosavljevic, Mladen Veinovic, and Branko Kovacevic. Estimation of nonstationary AR model using the weighted recursive least square algorithm. In *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, volume 2, pages 1432–1435. IEEE, 1995.

[29] Krishna Malladi and Ratnam Rajakumar. Estimation of time-varying AR models of speech through gauss-markov modeling. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 6, pages VI–305. IEEE, 2003.

[30] Sadaoki Furui. On the role of spectral transition for speech perception. *Journal of Acoustic Society of America*, 80(4):1016–1025, October 1986.

[31] James Jenkins, Winifred Strange, and Thomas Edman. Identification of vowels in vowelless syllables. *Perception and Psychophysics*, 34(5):441–450, September 1983.

[32] James W Pitton, Kuansan Wang, and Biing-Hwang Juang. Time-frequency analysis and auditory modeling for automatic recognition of speech. *Proceedings of the IEEE*, 84(9):1199–1215, 1996.

[33] Yves Grenier. Time varying lattices and autoregressive models: Parameter estimation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.*, volume 7, pages 1337–1340. IEEE, 1982.

[34] Marie Christine Chevalier, Gerard Chollet, and Yves Grenier. Speech analysis and restitution using time-depedent autoregressive models. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'85.*, volume 10, pages 501–504. IEEE, 1985.

[35] Yves Grenier and Marie Christine Omnes-Chevalier. Autoregressive models with time-dependent log area ratios. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(10):1602–1612, 1988.

[36] Panbong Ha and Souguil Ann. Robust time-varying parametric modelling of voiced speech. *Signal processing*, 42(3):311–317, 1995.

[37] Marko Juntunen and Jari Kaipio. Stabilization of TVAR models: A regularization approach. *University of Kuopio Department of Applied Physics Report Series*, (2/99), 1999.

[38] Jari Kaipio and Marko Juntunen. Deterministic regression smoothness priors TVAR modelling. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 3, pages 1693–1696. IEEE, 1999.

[39] Daniel Rudoy and Tryphon Georgiou. Regularized parametric models of nonstationary processes. In *Proc. the 19th International Symposium on Mathematical Theory of Networks and Systems (MTNS2010)*, pages 5–9, 2010.

[40] Srikanth Raj Chetupalli and Thippur Sreenivas. Time varying linear prediction using sparsity constraints. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6290–6293. IEEE, 2014.

[41] Louis Liporace. Linear estimation of nonstationary signals. *The Journal of the Acoustical Society of America*, 58(6):1288–1295, 1975.

[42] Mark Hall, Alan Oppenheim, and Alan Willsky. Time-varying parametric modeling of speech. *Signal Processing*, 5(3):267–285, 1983.

[43] Alan Wrench and Colin Cowan. A new approach to noise-robust LPC. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.*, volume 12, pages 305–307. IEEE, 1987.

[44] Cuntai Guan, Yongbin Chen, and Boziu Wu. Direct modulation on LPC coefficients with application to speech enhancement and improving the performance of speech recognition in noise. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 2, pages 107–110 vol.2, April 1993. doi: 10.1109/ICASSP.1993.319242.

[45] Fumitada Itakura. Minimum prediction residual principle applied to speech recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 23(1):67–72, Feb 1975. ISSN 0096-3518. doi: 10.1109/TASSP.1975.1162641.

[46] Bishnu Atal. Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification. *the Journal of the Acoustical Society of America*, 55(6):1304–1312, 1974.

[47] Karl Schnell and Arild Lacroix. Time-varying linear prediction for speech analysis and synthesis. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 3941–3944. IEEE, 2008.

[48] Trond Skogstad and Torbjørn Svendsen. Time-varying cepstral coefficients. *ISCA ITRW on Speech Analysis and Processing for Knowledge Discovery, Aalborg, Denmark*, 2008.

[49] Zafrir Babin, Felix Flomen, and Ilan Shallom. Incorporation of time varying AR modeling in speech recognition system based on dynamic programming. In *Electrical and Electronics Engineers in Israel, 1991. Proceedings., 17th Convention of*, pages 289–292, Mar 1991. doi: 10.1109/EEIS.1991.217640.

[50] Krishna Nathan and Harvey Silverman. Time-varying feature selection and classification of unvoiced stop consonants. *Speech and Audio Processing, IEEE Transactions on*, 2(3):395–405, Jul 1994. ISSN 1063-6676. doi: 10.1109/89.294353.

List of Publications

Peer-reviewed Journal

[1] Federico Ang, Rowena Cristina Guevara, Yoshikazu Miyanaga, Rhandley Cajote, Joel Ilao, Michael Gringo Angelo Bayona, Ann Franchesca Laguna, "Open Domain Continuous Filipino Speech Recognition: Challenges and Baseline Experiments," IEICE Transactions on Information and Systems, Vol. E97-D, No. 9, pp. 2443-2452, Sep. 2014.

International Conference

[1] Federico Ang, Hiroshi Tsutsui, Yoshikazu Miyanaga, "Time-Varying LP Cepstral Features for Improved Isolated Word Speech Recognition," to appear in Proceedings of IEEE International Conference on Digital Signal Processing (DSP), Singapore, 21-24 July 2015.

[2] Federico Ang, Hiroshi Tsutsui, Yoshikazu Miyanaga, "Incorporation of Time-Varying LP Cepstral Features in HMM-Based Isolated Word Speech Recognition," in Proceedings of International Symposium on Signals, Circuits and Systems (ISSCS), Iasi, Romania, 9-10 July 2015.

[3] Federico Ang, Yoshikazu Miyanaga, Rowena Cristina Guevara, Rhandley Cajote, Michael Gringo Angelo Bayona, "Open Domain Continuous Filipino Speech Recognition with Code-Switching," in Proceedings of IEEE International Symposium on Circuits and Systems, pp. 2301-2304, Melbourne, Australia, 1-5 June 2014. [doi:10.1109/ISCAS.2014.6865631]