A Study of the Automatic Speech Recognition Process and Speaker Adaptation

by

Ian Stokes-Rees

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical Engineering

Waterloo, Ontario, Canada, 2000

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis considers the entire automated speech recognition process and presents a standardised approach to LVCSR experimentation with HMMs. It also discusses various approaches to speaker adaptation such as MLLR and multiscale, and presents experimental results for cross-task speaker adaptation. An analysis of training parameters and data sufficiency for reasonable system performance estimates are also included.

It is found that Maximum Likelihood Linear Regression (MLLR) supervised adaptation can result in 6% reduction (absolute) in word error rate given only one minute of adaptation data, as compared with an unadapted model set trained on a different task. The unadapted system performed at 24% WER and the adapted system at 18% WER. This is achieved with only 4 to 7 adaptation classes per speaker, as generated from a regression tree.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis reviews the speech recognition problem and presents the current alternatives available for adaptation in Automatic Speech Recognition (ASR) systems. Recognition experiments are performed using Switchboard and Macrophone corpus'. A systematic experimental methodology is presented and applied to cross-task recognition.

There are four general goals for this dissertation: 1) provide a conceptual background of the field of speech processing, and speech recognition specifically; 2) provide a theoretical background for adaptation techniques applied to parametric acoustic modeling; 3) present and discuss experimental results for cross task adaptation; and, 4) establish a baseline experimental approach for ASR research.

1.1 Objectives of ASR

Speech recognition aims to provide a natural and efficient mode of interface for transferring information from humans to computers. In many ways it is felt that speech is the most natural form of communication for humans, and therefore it is desirable to use this natural mode when communicating with automated devices. Currently, the keyboard and mouse are the primary forms of input to a computer, and a monitor and printer are the primary forms of output. While these devices have their advantages in terms of lack of ambiguity, the physical space they occupy limits their use. A speech interface would reduce the physical size of the interface package to a

microphone and speaker. Current systems are able to perform speaker dependent recognition in a controlled environment to very high degrees of accuracy and therefore usability, but the goal of speaker independent recognition and robustness to environmental noise or variation in recording quality has yet to be accomplished.

The basic measure of system performance is word error rate (WER) which combines deletions, substitutions, and insertions. The best speaker dependent systems such as IBM Via Voice or Dragon Naturally Speaking provide less than 1% WER once trained for a particular speaker, with a standard accent, in a good recording environment and with a high quality microphone and digital sampling rate. In comparison, the benchmark of general continuous conversation speech recognition in a speaker independent environment is the Switchboard corpus for which results lower than 40% WER have yet to be achieved.[7] It is generally considered that performance of better than 5% WER is required for a system to be useful, otherwise the effort required for correcting errors out weighs the benefits. For common usage, it is believed that performance better than 1% WER is required.

The continuing process of experimentation and refinement of ASR systems will hopefully be able to produce human or super-human recognition systems in time. The following is a list of the characteristics of an ideal ASR system:

- speaker independent (rapid adaptation, speaker recognition, and gender detection)
- language independent (language recognition)
- rate independent
- resilient to noise (noise rejection, or filtering)
- correct inference from ambiguity (e.g. you're vs. your)
- stream selection (multiple simultaneous speakers or multipath)
- intelligent (able to learn new grammar, language, semantics, syntax, dialect)

While current systems are functional, they lack most of the ideal characteristics listed above. Research systems which process acoustical data off-line allow theoretically unlimited computing resource to tackle the ASR problem and do not perform better than 70% correct when processing

conversational speech in a speaker independent (SI) system even with a high SNR. While this has discouraged researchers, commercial speaker dependent (SD) systems are able to achieve 99% accuracy in a favourable recording environment with a good quality microphone and a "standard" dialect of the language for which the system is intended. This suggests the weakness lies in the speaker independent model, rather than lack of computing power.

1.2 Acoustic Adaptation

A current area of great interest is the adaptation of acoustic models. It is hoped that adaptation of various forms can compensate for the poor performance of current SI ASR systems which generally use static stochastic models for the acoustics (i.e. a fixed statistical distribution for each model). Such adaptation may be able to take into account changes in environment, noise, dialect, and speaker.

This thesis specifically reviews Maximum Likelihood Linear Regression (MLLR), and Multi-scale (MS) adaptation, and experimental results are presented for cross-task MLLR adaptation. Adaptation has generally been applied for speaker adaptation, and task adaptation is a new area with different problems associated with it.

Specifically, the 10hr subset of the Switchboard corpus is used as the base training model, with recognition (testing) performed using the Macrophone corpus. Various approaches are used for adapting the Switchboard models to maximize recognition performance on the Macrophone data.

1.3 Overview

This thesis discusses the current state of the art Automatic Speech Recognition (ASR) technology, and presents experimental results for speaker independent Large Vocabulary Continuous Speech Recognition (LVCSR), currently the most demanding task in ASR. Chapter 1 provides a brief background and overview of the contents. Chapters 2 presents a conceptual background and history of Speech Processing, and the various approaches to ASR. Chapter 3 present a theoretical background in Hidden Markov Models (HMMs) and model adaptation techniques. Chapter 4 presents and discusses experimental results for cross-task recognition using Switchboard and

Macrophone. Chapter 5 presents and defends a particular standardized experimental methodology for the development of HMM-based ASR systems. Chapter 6 summarizes experimental findings, and discusses future research work in ASR and adaptation.

Chapter 2

A Background In Speech Processing

2.1 The Goal

The exponential decrease in computing cost coupled with the exponential increase in computing power has made continuous speech recognition systems both possible and affordable. The next several years will likely see rapid growth in the demand for ASR systems and possibly a paradigm shift in computing interaction to speech centered interfaces. This shift may be the most dramatic yet in the chain that has gone from the punch-card, to the keyboard, to the mouse, and now to voice in improving the mode by which humans interact with computers.

Broadly speaking, speech is a means for communicating information. The human brain makes use of the articulatory system to communicate ideas through acoustic waves. These acoustic waves can then be transmitted by various means to other people, allowing the auditory system to then decode these waves allowing the receipt of the original idea. Possessing a complex communication system which is learned and not genetic and the ability to receive and assimilate complex ideas through that system are critical parts of human learning and therefore cognition. An autonomous model of the human communication system is, therefore, a step towards understanding human cognition.

Furthermore, developing an autonomous (i.e. computer based) equivalent to the human communication system provides the opportunity for an improved Human-Computer Interface (HCI). Many computerized systems are designed with the objective of providing tools to either increase productivity of the users or to improve their accessibility to information. The progression from punch-card, to keyboard, to mouse have provided increasingly natural interface modes. The ability to provide speech as an alternative for HCI would be a great jump in naturalizing interfaces.[29]

Although reproducing the human communication system does not imply an understanding of the process it does provide insight into the nuances and complexities of the process and therefore a greater understanding of language, learning, and cognition. Effective models of human communication can be used to improve HCI through both speech generation for output and speech recognition for input.

2.2 Fundamental Equation of Speech Recognition

The problem of speech recognition has been succinctly described by a single equation which captures all the elements of a typical speech recognition system. Equation 2.1 implies a numeric parameterization of acoustic information, a prior model for the acoustics, and a prior model for the language.

$$\begin{aligned} O &= o_1, o_2, \ldots, o_T \\ arg_i max[P(w_i|O)] \\ P(w_i|O) &= \frac{P(O|w_i) \cdot P(w_i)}{P(O)} \end{aligned} \tag{2.1}$$

In equation 2.1 O represents a sequence of T observations, while $arg_i max$ specifies the selection of word i as the highest probability word given the observation sequence O. This equation can be rearranged by Bayes Rule such that it is a combination of the acoustic model $P(O|w_i)$ and the language model $P(w_i)$. The probability of the acoustic sequence does not need to be computed since it is simply a scaling factor, independent of the maximization function to select the most probable word $w_{i,best}$.[1]

The acoustic model structure is not specified by this equation, but is a stochastic function which will provide a probability of the observation sequence having been generated by it. Likewise, the

language model specifies the likelihood of the word occurring. This particular formulation of the fundamental equation of speech recognition takes into account only single word recognition. Multiword recognition would likely include a language model which accounted for the context of the word, while the acoustic model would be faced with the problem of determining word boundaries. Further discussion of language models is limited to that contained later in this chapter, while issues around acoustic models are covered more fully in chapter 3.

2.3 Acoustic Models

Concepts of acoustic resonance have been recognized for thousands of years as is attested to by the existence of musical instruments throughout human civilization. The same laws of physics which describe tonal patterns in musical instruments also apply to human acoustics. Generally a computational model of the vocal tract shows a circuit like model with a driving source (the diaphragm and lungs), and a Y circuit where the pharyngeal cavity branches into the nasal and oral cavities. Impedance variations due to changing cavity size create variations in the base standing wave which is produced between either a fixed node at the source and an open or closed node at the lips or nostrils.

This driven impedance circuit model with a standing wave is identical to transmission line models for wave propagation through electrical circuits. Electrical models can be applied directly to the acoustic model to determine energy transfer functions and predict resulting output wave characteristics in terms of frequency and amplitude. A further benefit of this representation of the articulation process is in the field of speech generation. The transmission model of the vocal tract can be represented using a transfer function. The various parameters of the transfer function can be associated with sound specific characteristics or speaker specific characteristics, as well as variations in the driving function which is the source wave produced by the lungs, diaphragm, and vocal chords.[9] This model attempts to quantize the effects of air mass dynamics, friction, tissue and compressibility.

Figure 2.1 illustrates a schematic of the speech production system along with an equivalent T circuit model for each of the major resonance cavities. In [9] Flanagan attributes the four components R, G, C, and L to specific physical phenomena relating to the propagation of an air mass representing a longitudinal wave moving through the vocal tract. The calculations for these

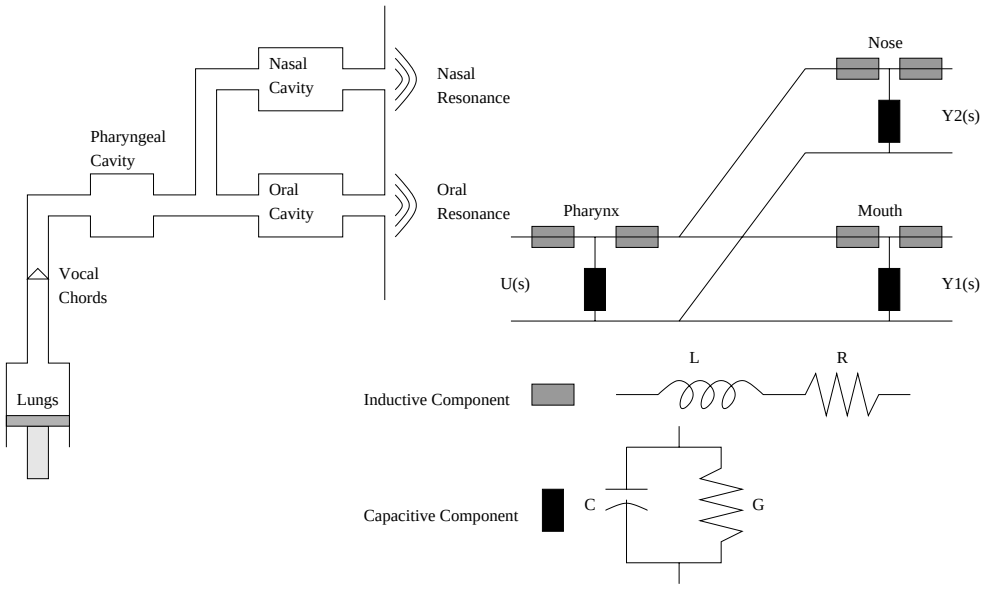

Figure 2.1: Schematic and circuit model of the vocal tract [9]

quantities is derived from quantities such as thermal conductivity, viscosity coefficients, and air density. L refers to the acoustic inductance of the moving air mass, since the air mass will create a suction effect drawing air behind it as it passes through the vocal tract. C refers to the acoustic capacitance which is the stored energy as the air mass experiences compression and deflation. R is the frictional losses of the air mass through viscosity with the flesh walls of the vocal tract. G represents the power loss through heat conduction at the walls.

Research during the 1960s and 1970s aimed at achieving advances in speech recognition through a better understanding of the speech system. This kind of circuit model, while useful for analytic speech production models and a general understanding of human physiology, have not been beneficial in the task of automated recognition.

This class of speech production model is called the source-filter model, since it describes two basic components: the source driving function $u(t)$; and a transfer function $h(t)$. Together these produce the perceived output given by the equation $u(t) \otimes h(t) = y(t)$. This model has several advantages, one being the all-pole model simplification. Given that the transfer function of a lossless tube can be described by an all-pole model it is possible to approximate the vocal tract by:

$$H(z) = \frac{1}{1 - \sum_{i=1}^{p} a_i z^{-i}} \tag{2.2}$$

In equation 2.2, z^{-i} represents the delay operator. The coefficients a_i can be estimated by minimum mean squared error (MMSE) techniques and these values form what are known as the Linear Predictive Code (LPC) of the speech signal. This all-pole parameterization uses a fixed number p of previous samples to find the best fit for an auto-regressive predictor. While this technique has been popular for sometime, it has significant shortcomings when coping with nasal or fricative sounds – one does not fit an all-pole assumption, and the other has the driving source at the output (lips) rather than lungs. As well, LPC is sensitive to pitch rate and noise. These are key reasons MFCC is now the preferred parameterization format for speech processing.

2.4 Signal Parameterization

Early approaches of parameterizing speech through the use of binned frequency components has evolved into the mel frequency cepstrum coefficients (MFCC). This is the most popular form of parameterization for speech recognition as it seems to provide good separation of spectral acoustic components, noise rejection, and a degree of normalization. The mel scale, which is used for determining the frequency bin size and spacing, takes into account the nonlinear sensitivity of the human ear to various frequency components, and the spectral distribution of speech information. In simple terms, the human ear produces a linear response up to 1 kHz and a logarithmic response beyond this.

$$m = 1125 * \log(0.0016 * f + 1) \tag{2.3}$$

Equation 2.3 is the mapping of frequency to mel scale. Coefficients are generated by approximately uniform sized bins from this mel scale. An advantage of this approach is that it allows narrow band representation of LF components which is where harmonics (formants) are usually found, while still retaining good temporal resolution at HF so short acoustic events can be detected (such as bursts). Figure 2.2 shows a typical binning for an 11 parameter mel scale filter.

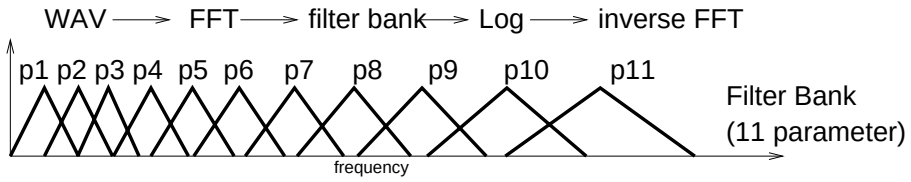

Figure 2.2: Bin filters using mel scale

The cepstrum is a spectral transformation on the acoustic vector (the name derives from a rearrangement of the word spectrum) generated by an FFT, which takes the log of the frequency and then an inverse FFT. If human speech is thought to consist of several independent time varying components each convoluted together then the inverse FFT of the log of the frequency domain signal amounts to a deconvolution such that the various components add together in the cepstrum domain. MFCC parameterization was used in all experiments discussed in this thesis.

Furthermore, common practice is to use both windowing and signal pre-emphasis in the parameterization process to improve the signal representation. The objective is to retain as much acoustic information as possible from the original time varying speech signal. To avoid high frequency noise resulting from square (i.e. cutoff) windowing before the FFT, a Hamming window is used. This window has a peak at the mid point and conserves the overall signal power, while tapering the signal strength to zero at the edges. The pre-emphasis process boosts the high frequency signal since most of the spectral power is found at low frequencies and this will bias the recognition process in favor of matching LF coefficients over HF ones, even though experience shows that substantial acoustic information is contained in the relatively weak HF signal.

Using a sample rate of 10 ms (100 Hz) provides fairly stationary parameterization, and to improve signal estimates the samples are calculated from overlapped windows of size 20-30 ms. Since FFTs are typically required in this process the base signal sampling rate in the time domain needs to be at least 8 kHz, if not more. This is to satisfy the Nyquist criterion of twice the signal bandwidth, meaning that 8kHz would provide up to 4 kHz bandwidth. While most of the relevant acoustic information in adult speech is contained below 4 kHz, the human audible range reaches as high as 20 kHz, implying that up to 40 kHz sample rate may be required to capture information at all frequencies.

In summary, the parameterization of a speech amplitude signal (one dimensional) is transformed to a vector representation which contains both temporal and frequency information through

the use of a moving window FFT, coefficient binning, and pre-emphasis. These vectors typically represent 10ms of speech, which has been shown to be sufficiently short for the acoustics to be considered stationary. Energy and first and second derivatives of the coefficients may also be included in the transformed sample vector to provide additional signal information pertaining to the volume and acoustic dynamics.

2.5 Acoustic Units

Extensive research into human speech by phonologists has produced the International Phonetic Association (IPA) alphabet which represents most sounds which can be produced by humans in terms of phonemes. These are taken to be the smallest acoustic unit of speech. The actual realization of these phonemes in speech are referred to as phones and due to the smooth and continuous nature of speech are not clearly defined (i.e. one phone will usually blend into the next). Convention has phonemes represented by square brackets and phones by hashes (e.g. the phoneme [*ah*] or the phone /*ah*/).

Table 2.1 lists the TIMIT phone set used for the experiments discussed in this thesis. It is a subset of the ARPAbet developed by ARPA for use in speech recognition trials to ease the integration of phonetic symbols in an ASCII computing environment. It is important to recognize that both smaller and larger acoustic units may be used, however the reasonably compact size of the phone set and its diverse ability to represent speech events make it the unit of choice among researchers today. The TIMIT set was chosen by Texas Instruments and MIT for its compactness and good representation of phones occurring in American English.

While table 2.1 lists the phonemes sorted alphabetically with examples, a much more revealing organization is shown in figure 2.3 where the TIMIT phone set is arranged on a hierarchical tree. This shows just one way of grouping phones – a process which will be discussed further in Chapter 3 and Chapter 4 since it is critical to the adaptation process. Inspecting figure 2.3 also reveals the contradictory discrimination processes which are required in speech processing. On a coarse level the various phone groups have distinct acoustic differences and therefore a pattern recognition system must be able to identify the gross characteristics of, for example, the

Phone	Example
AA	father
AE	had
AH	mud
AO	fought
AW	how
AY	hide
B	bat
CH	church
D	deep
DH	then
EH	head
ER	heard
EY	hay
F	five
G	go
HH	help
IH	hid
IY	heed
JH	just

Phone	Example
K	kick
L	love
M	mom
N	noon
NG	sing
OW	hotel
OY	boy
P	pea
R	race
S	so
SH	show
T	tea
TH	thing
UH	hood
UW	who
V	vice
W	want
Y	yard
Z	zebra
ZH	measure

Table 2.1: TIMIT phone set

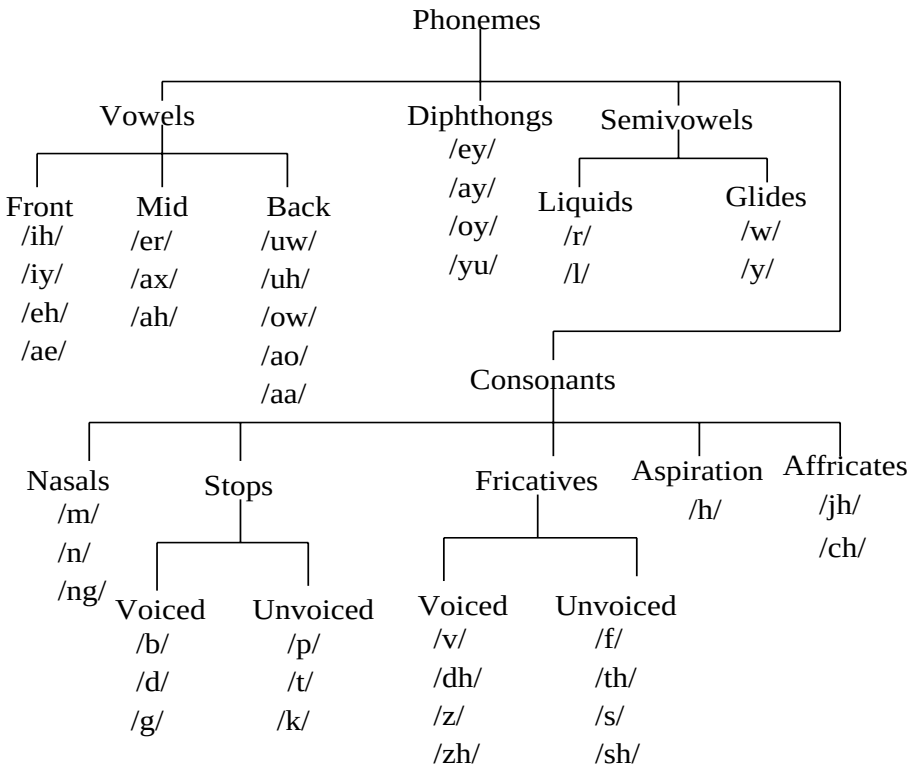

Figure 2.3: Tree classification of TIMIT phone set

nasals vs. the fricatives. However, on a fine scale, the same system must be able to identify the nuances between similar sounds such as the nasals /m/ and /n/. Yet another dimension of the opposing challenges faced by recognition systems is that of duration, where some acoustic events are characterized by the average over 30 to 60 ms, and others occur in less that 10 ms. As an example, vowels typically last 30 to 60 ms and any spurious signal during that time can safely be taken to represent unimportant noise. On the other hand, stops such as /d/ or /t/ occur very quickly and are easily missed by the averaging, filtered, windowed parameterization process.

Syllable units are also frequently used in speech processing, however there are over 400 syllables in English and the apparent necessity of including contextual models make context-sensitive syllable models unreasonable. This would imply an upper limit of 64 million models in a tri-syllable context, compared to 125 thousand tri-phones, which is currently the more popular context based approach. For simple tasks with small language models it may be reasonable to use word-based models. This is common for digit or character recognition applications.

Context models have proven to be a critical development in the improvement of speech recognition systems. While early models used single phone or syllable models (i.e. around 50 or 400 base models respectively), experiments involving the use of context based models proved the importance of the adjacent acoustic information, since coarticulation occurs for many phonetic sequences. Context models can be dependent on the left or right acoustic models, such that a phone sequence "/a/ /b/ /c/ /d/" would become "/**a**/ /**a**+b/ /a-**b**/ /**b**+c/ /b-**c**/ /**c**+d/ /c-**d**/ /**d**/". The bold face phone represents the base phone, but - or + signify left (prior phone) or right (subsequent phone) context for that base phone. In this way a base phone /p/ would have several left and right context phones /l-p/ and /p+r/, each which attempted to captured the subtle differences that the left phone /l/ or right phone /r/ had on the base phone. Given a set of 50 base phones, this would provide an upper limit of 50*50 left plus 50*50 right + 50 base phones for a total of 5050 models. In practice, not all of these phones would occur in natural speech or the phonetic dictionary. A further context model, the triphone /l-p+r/, includes both the left and right phones in each context model, so the upper phone limit now also must include 50*50*50 for over 130 thousand models. Again, practical limitations on which phones actually occur limits this to around 20 thousand. The triphone is the most popular context model today, although work has been done using quinephone (5) or heptaphone (7) context model.

2.6 Parametric Models of Acoustic Units

There are various ways of representing the acoustic models. Due to computational limitations, early models prior to the late 1980s used vector quantization where each model was represented by a single vector or sequence of vectors.[12] Using some distance metric new speech vectors can be recognized by finding the closest match with the known models. These recognized vectors can be used to update the original model vector. Euclidean distance measures are relatively easy to compute and the model is represented by a small number of parameters corresponding to the mean value for each coefficient from the training data.

These models are too rigid to cope with the acoustic variation which occurs in natural speech and therefore were followed by statistical models in the late 1980s and early 1990s when it became feasible to apply statistical signal processing to the speech recognition problem (although it should be noted that CMU and other institutions had implemented statistical speech recognition systems

in the 1970s). The advantage of statistical models is that they provide powerful mathematical models for continuous random signals, and therefore more closely match the characteristics of speech than a deterministic model such as vector quantization.

The most popular statistical model to date is the Hidden Markov Model. The background for markov models and markov chains was developed by Baum in the 1960s [20] [5], but it was not until the 1970s that this approach was applied to speech recognition by Baker at CMU [4] and Jelinek at IBM [14], and not until the 1980s that it was widely adopted and recognized as the preeminent modeling approach. Markov chains make a simplifying assumption that observations are independent conditional on only the last set of observations. Provided a proper formulation of the problem, this assumption, while not strictly true, does produce very accurate modeling results for speech.

Using HMMs, each acoustic unit typically has it's own model. The model represents a statistical description of the sample vector characteristics. Since these models generally have several states, each state can have it's own vector description, thereby providing a means of representing the sample vector dynamics with time. This is important since it is well known that only vowels can be easily approximated as stationary over their duration, and even then co-articulation usually implies variation at the phone onset and finish. The coefficient statistical distribution usually is defined as Gaussian, although work has been done in making use of other distributions with HMMs [18]. Using the EM algorithm, it is possible to approximate general statistical distributions using a mixture of multiple Gaussian. Further discussion of the HMM is saved for Chapter 3 since the HMM has been used as the base model for adaptation in this thesis.

2.7 Language Models

The language model has been a source of significant difficulty in speech processing since the outset. While it is generally accepted that significant information is contained in the *a priori* knowledge of syntax and semantics, it has not yet been successfully transfered to an automated system. As a result, statistical language models based on word pairs or word sequences have provided the best results to date. The ideal language model would be sufficiently flexible to accept a random sequence of words, and yet able to predict missing or noise corrupted words and sounds based on the context. Typical language model implementations do not dynamically adjust the influence of

the LM on the recognition process and therefore cannot do this. Their primary responsibility is to limit the search space for model sequences.

The dictionary is a critical component of the language model and provides a selection of all possible phonetic "spellings" of the words contained in the dictionary. Dictionary size is an important issue both in terms of the possible size of the recognition vocabulary and the decoding complexity which increases with the number of words. An important balance must be achieved between the richness of the available vocabulary and the possibility for confusion. Since 80% of common conversational speech consists of a vocabulary of 2000 words or less, it may seem that the recognition task should simply focus on this core vocabulary. However, considering information theory and the concept of symbol entropy it is clear that the less frequently a word occurs the more information there is likely to be associated with it. For that reason the 2000 common words largely do not carry the information component of speech, and it is important to be able to recognize the uncommon words well.

Since the LM typically is not formed using *a priori* knowledge of the language syntax and semantics, it can be based on either a simple word loop where each word is equally likely to occur at all times (a valid model for digit or command recognition) or using some training texts to generate word sequence likelihoods.

Chapter 3

Theoretical Background

An overview of the theoretical techniques is required to discuss the application of adaptation to speech recognition. This chapter discusses the most common model used in speech recognition, a continuous density Hidden Markov Model (HMM), in terms of it's form and features which suit it to the task. Following this, a discussion of Maximum Likelihood Linear Regression (MLLR) models presents the mechanics of this adaptation approach, and the key issues involved in it's implementation. Finally the technique of multi-scale adaptation is presented.

3.1 The Hidden Markov Model

As discussed in chapter 2, the HMM has become the standard method of acoustic modeling used in speech recognition. This is due to the rigorous and well understood mathematical background for the model, and its ease of integration with stochastic approaches. There are three fundamental design problems to be solved when using HMMs: i) evaluating a sequence probability given a specific model; ii) choosing the optimal model state sequence; and, iii) adjusting the models to best fit the specified data.

The term "hidden" refers to the fact that each model consists of several states, and observations are a probabilistic function of the particular state, which is unknown. Referring to the fundamental equation of speech recognition, 2.1, this amounts to a transformation of the acoustic

model term $P(O|w_i)$ to $P(O, X|M)$.[21] That is, instead of calculating the likelihood of an observation sequence O based on a particular word, an estimate of a state sequence X which generated the observations is made, given a set of models M which parametrically describe each state. It is now this state sequence which is unknown, hence the term "hidden". In this new context we have a likelihood of observing each observation o_t given a particular state j. This term is indicated by $b_j(o_t)$. A state transition model a_{ij} describes the likelihood of transition from state i to state j. Now the acoustic model term from the fundamental equation can be restated as:

$$P(O, X|M) = \pi_{s_1} b_{s_1}(o_1) a_{s_1 s_2} b_{s_2}(o_2) a_{s_2 s_3} b_{s_3}(o_3) \ldots a_{s_{i-1} s_i} b_{s_i}(o_i) \tag{3.1}$$

This equation introduces the final component of the HMM equation which is the initialization problem for selection of the first state π_{s_1}. At each time index i a different state is possible, therefore there is always a transition likelihood $a_{s_{i-1} s_i}$ from $i-1$ to i as well as the observation likelihood $b_{s_i}(o_i)$. In this way the entire collection of HMMs are characterized by a parameter set $[A, B, \Pi]$.

The problem of calculating the three components $\pi_{s_1}, a_{s_{i-1,i}}, b_{s_i}(o_i)$ is not difficult given that a set of training data is supplied. The process involves an initialization then an iterative re-estimation of the components through the EM algorithm. The challenge is in data sufficiency for accurate estimates. Algorithms to perform this estimation were established by Baum in [20] and [5]. They describe the EM algorithm for first order markov chains, in which observations are independent conditioned only on the state which generated them – this is the first order markov approximation.

Considering the estimates for the number of models which typically exist in practical speech recognition systems (discussed in chapter 2 to be in the vicinity of tens of thousands) it may, at first glance, seem intractable processing the $1e4^2 = 1e8$ possible state transitions for all t time intervals thus suggesting $1e8^t$ possible transitions overall and complexity of $O(N^t)$. A limiting factor comes from the language model which asserts a very limited word internal state transition structure and a fixed number of possible inter-word transitions. The primary limiting factor was recognized by Viterbi in his lattice decoding algorithm which greatly simplified this otherwise intractable problem. If one imagines a grid where the horizontal axis represents time and the vertical axis represents all possible states, then the state sequence is simply a path through this

grid. The challenge is in selecting the optimal path. The complexity, in fact, is not $O(N^t)$ but only $O(N^2)$ in the worst case since the Viterbi algorithm [23] allows lattice decoding using only the last set of all possible state transitions (N states each transiting to a maximum of N next states for $O(N^2)$).

Typical models make use of a low order number of states to represent each phone. Three is a common value, since this has heuristic appeal considering a structure of onset, intermediate, and exit regions for each phone. This is unrelated to the concept of the tri-phone which creates a context dependent model based on adjacent phones, although given a 3 state tri-phone model $/l-p+r/$ with states $s_1, s_2, s3$, it is reasonable to hypothesize that s_1 could be close to the terminal state of a "typical" $/l/$ model, and s_3 could be close to the entry state of a "typical" $/r/$ model.

Gaussians are typically used for modeling the parameters, as they are easily calculated and contain only two parameters. Other stochastic models have been presented [18], but the simplicity and success of Gaussians has meant that they have remained the most common stochastic parameter model. While it is reasonable to assume that the statistics for a particular parameter of a particular acoustic model for a single speaker would likely have a Gaussian distribution, speaker variation in the form of dialect, or gender and age related pitch variances are not well represented by a single mean and variance. It has become popular to approximate general distribution functions through the use of a mixture of Gaussians. Training multi-mixture models, as they are called, is done by successively splitting models and retraining using the Baum-Welch algorithm.

This approach has been extended even so far as to eliminate multiple models and simply group all models together with a large number of mixtures. This approach selects acoustic units based on the closest matching mixture. In practice this has not been successful in improving recognition results as it eliminates the valuable lattice decoding process which allows an additional level of structure to be asserted in the model. Typical systems use 3 to 7 mixtures per state which seem to provide adequate coverage of gender, age, and dialect modalities.

The above suggestion concerning the cross-model association of states has been extensively analyzed and it is common practice to perform state-tying, where a meta-state is defined which is shared between several models. While attempts at performing this tying heuristically have been attempted, the greatest improvements have always resulted from the use of numerically determined tying.

3.1.1 Adaptation of HMMs

Adaptation of HMMs has two distinct components. One is the adaptation which is done in the iterative re-training process. This is carried out by the EM algorithm through the methods established in what is known as the Baum-Welch algorithm. This process iteratively alternates between using the existing models to estimate the state sequence, then maximizing the model parameters to best match the observations assuming the state sequence is correct. Provided this process is initialized properly it will tend towards improved state sequence estimates which will thereby provide improved parameter estimates, and so on. While this can be described as model adaptation, this process is simply called "training".

The adaptation process makes use of a model which is considered "trained" or at least "well initialized", then adjusts it for the current operating circumstances. This usually means making use of small amounts of data, in comparison to the data set used for training, to adjust parameters to capture the characteristics of the adaptation data. Adaptation can consist of adjusting parameters, shifting mixture weights, changing state tying, or synthesizing new language model pronounciations. Due to the limited amount of data available for adaptation, almost by definition of the process, it is necessary to have some mechanism by which observations for certain models and states can be shared more generally to provide useful adaptation of a large part of, if not the entire, model set.

Some basic generalizations use only the center phone of the adaptation data to generate a transformation function for the entire set of states sharing the same center phone. More advanced adaptation models use tree structures to assert a hierarchical structure to model associations. Both MLLR and multiscale use tree structures for representing adaptation correlations. The primary hurdle of most adaptation approaches is to overcome the independence assumption asserted by most acoustic models. This assumption makes the speech recognition problem tractable, such that the overall model probability is equivalent to the product of the probability of each model. This is shown by equation 3.2.

$$p(\theta) = \prod_{i=0}^{L} p(\theta_i) \tag{3.2}$$

When the model independence is asserted in this way, it is difficult to share adaptation data between different models. The adaptation problem then becomes the approximation of some cor-

relation structure between models which can be used to share data and adapt correlated models. The following two sections discuss two model based adaptation approaches. As well as adapting the acoustic models, it is possible to perform source adaptation, such as vocal tract length normalization which attempts to normalize the pitch, and dynamic time warping which attempts to compensate for varying speaker rates and state distributions. Neither of these approaches will be discussed further here.

3.2 Maximum Likelihood Linear Regression

Initially proposed by Leggetter and Woodland in [25] and [26], this approach performs a linear transform on Gaussian models. The most general linear transform of a Gaussian is given by equation 3.3.

$$\mathcal{N}(\mu, \Sigma) \rightarrow \mathcal{N}(A\mu + b, A\Sigma A^T) = \mathcal{N}(\hat{\mu}, \hat{\Sigma}) \tag{3.3}$$

Empirical results suggest that covariance adaptation is secondary to the benefits gained from mean adaptation, therefore the transformation $A\Sigma A^T$ is usually not performed. Since the transformation matrix A already has been estimated for the mean transformation, this step is primarily skipped for computation reasons, although it can be argued that if A is used to adapt Σ then adaptation data covariance should be used to estimate the transform A, which significantly increases the complexity of the estimation problem for the matrix A.

Consider a mean vector μ of size n, then A is an $n \times n$ square matrix and b is an $n \times 1$ vector. The transform matrix A performs scaling and linear recombination of the mean elements, allowing cross parameter adaptation, while the vector b adds a bias component. A and b can be merged into a single extended adaptation matrix W of size $n \times n+1$ where W has the form shown in equation 3.4. This also shows the necessary transform on the mean vector μ which has a one concatenated on the end. It is plain to see that the new transform $W\hat{\mu}$ is exactly equivalent to $A\mu + b$.

$$W = \begin{bmatrix} A & b \end{bmatrix}$$
$$\hat{\mu} = \begin{bmatrix} \mu \\ 1 \end{bmatrix} \tag{3.4}$$

Considering a single Gaussian pdf for the state s with observation set o, the pdf can be expressed as:

$$b_s(\mathbf{o}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{o}-W\hat{\mu})^T\Sigma^{-1}(\mathbf{o}-W\hat{\mu})} \tag{3.5}$$

Assuming that a well trained base model already exists, it is possible to use supplied transcriptions to determine initial state occupation, leaving only the estimate of W, which is made by maximizing the likelihood of the new model $\mathcal{N}(W\mu, \Sigma)$ matching the adaptation data $\mathbf{o}$. Details of this algorithm are given in [26].

A typical problem of acoustic adaptation is data sparsity. For adequate coverage of tens of thousands of states, upwards of an hour of data is required to produce reasonable statistical estimates, even if those estimates are only for adaptation purposes, using well trained base models. To overcome this, sharing of adaptation data between various states is crucial. MLLR does not specify a mechanism by which to group adaptation transforms, so various approaches have been attempted.[24] The trade off is between maximizing the number of adaptation classes to capture speaker dependent characteristics, and estimating the adaptation transforms adequately. Adaptation classes are therefore generated heuristically or using some rule based system that splits all states into successive subgroups until further splitting would exceed some threshold, such as minimum number of observations. Some approaches have looked at speaker clustering to share adaptation data between speakers with similar characteristics. [27]

Once regression classes have been defined, the observations for all states in a particular class are used to estimate the adaptation transform for that class, and that transform is then used to adapt all states in the class. Heuristic approaches use structures such as the phone tree in Figure 2.3 to group states based on their center phone and phonetic class. Other approaches use Euclidean distance measures of the mean vectors to perform kNN clustering of states[3], or use the structure established from a decision tree[2].

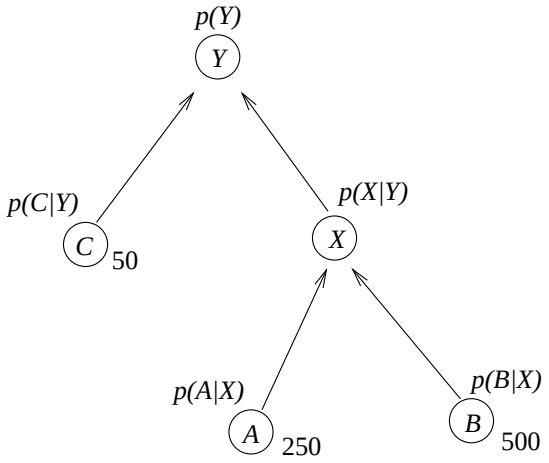

Figure 3.1: Simplified dependence tree

3.2.1 Hierarchical Adaptation Using Tree Structures

Decision trees use a set of context questions to create a binary tree which segments the entire state space hierarchically. Starting with all states in the root node, the set of context questions is applied. Every question has a true or false answer, thereby producing a particular binary division of all states at that node, for that particular question: all states which answer true to the question in one child cluster, and all states answering false in another. With N decision tree questions there are N possible binary divisions of the node. The split which maximizes the likelihood of the resulting models is used.

Decision tree, kNN, and other state clustering approaches often have a tree structure which provides a clear mechanism for representing multiple levels of clustering – a cluster hierarchy. Although it is possible to define clusters based on available adaptation data, thereby assuring that sufficient data exists in each cluster, typically different clusters will have greatly varying amounts of adaptation data. Using a minimum data threshold for generating an adaptation class, it is possible to cluster data up a tree until the threshold is reached, and then estimate the adaptation transform. In this way, adaptation transforms exist at all nodes in the tree where there is sufficient data (either intrinsically or through grouping data from child nodes), rather than just at the leaf nodes where the state models are clustered.

Referring to a simple example illustrated in figure 3.1 with 3 leaf nodes A, B, C and 2 internal

nodes X, Y, where $X = A + B$ and $Y = A + B + C$, if the sample threshold was 400 then node B would have sufficient data to have its own adaptation transform matrix W_B estimated. Nodes A and C, however, are below the threshold and would therefore be transformed by a regression class generated at the earliest node with sufficient data. For node A, this would be node X which would accumulate data from nodes A and B to use 750 samples to estimate the transform matrix W_{A+B}. Similarly, node C would need to use the transform W_{A+B+C} from node Y, estimated from 800 samples accumulated from all the child nodes. [10] [11]

3.3 Multiscale Adaptation

The last two years have seen the development of a remarkable system for characterizing interclass similarities using a stochastic, hierarchical model. The work by Kannan [15] has begun the process of identifying the correlation structure between classes using statistical models. This approach seeks to model the interclass covariance matrix using a multi-scale model. With a correlation structure represented through a covariance matrix it is theoretically possible to map acoustic vectors from one class into their equivalent representation in another class through a basis transform.

Prior to this approach, once state-tying has been completed, the remaining states had been assumed to be independent, so that the probability density function for the parameters of the entire model set consisting of L states is given by:

$$p(\mathbf{\Theta}) = \prod_{i=1}^{L} p(\theta_i) \tag{3.6}$$

This implies that two very similar models which are not tied to share the same state cannot benefit from the estimate data associated with the other state. This has been recognized as a modeling shortcoming, and it is hoped that multi-scale models can overcome the problem.

It is regarded as an impossible task to estimate a covariance matrix for thousands of states directly, however it has been shown that a multi-scale tree process can be used to represent the correlation structure. [8] Chow and Liu [16] presented the dependence tree structure (not to be confused with a decision tree) which approximates an N dimension correlation structure through the product of $N - 1$ second order distributions as given in equation 3.7. This equation can be

seen to represent a general tree where $\pi(i)$ indicates the parent of class i. The interpretation, then, is that a class is conditionally independent, given the parent nodes class.

$$p(\mathbf{\Theta}) = \prod_{i=1}^{N-1} p(\theta_i|\theta_{\pi(i)}) \tag{3.7}$$

Referring to figure 3.1, if the entire model is made up of the individual classes, then $M = A, B, C, X, Y$, and the approximated probability density function is given by:

$$p(M) = p(Y) \cdot p(C|Y) \cdot p(X|Y) \cdot p(A|X) \cdot p(B|X) \tag{3.8}$$

It can also be seen that an interesting result of simplifying the tree structure to have at most one child per node produces a linear 1st order markov chain, which is exactly the structure of an HMM. In the tree form, the dependence structure can still be estimated using the same mathematical approach used with estimating HMMs. The variation is that tree scale (or depth) does not have a temporal association as with linear HMMs where successive states correspond to successive observations in time, but rather scale represents an associative hierarchy.

Chow and Liu present an information theory approach for generating an optimal tree topology. If $p = p(\mathbf{\Theta})$ represents the probability density function of the entire model set, and $p_a = p(\hat{\mathbf{\Theta}})$ represents some approximation of that distribution, then a similarity measure is $I(p, p_a)$ – the mutual information of p and p_a. If the mutual information is zero, then p_a is an exact representation of p, and $I(\cdot)$ is always positive as given by equation 3.9.

$$I(p, p_a) = \sum_{\theta} p(\mathbf{\Theta}) \cdot log \frac{p(\mathbf{\Theta})}{p_a(\mathbf{\Theta})} \tag{3.9}$$

Minimizing this value has the effect of maximizing the closeness of p_a to p. They go on to show that this criteria is equivalent to maximizing the dependence tree weight, which is the sum of the mutual information between any two nodes (classes), also known as the branch weight. The interpretation of this is that two nodes i and $\pi(i)$ are best selected when they are maximally related (i.e. contain the maximum mutual information) as given by $I(i, \pi(i))$, since node i should be conditioned upon the node $\pi(i)$ to which it is most similar when asserting its independence from all other nodes. This has the objective of minimizing the effect of asserting first-order

conditional independence. A tree generating algorithm is presented which guarantees maximum tree weight, therefore minimum distribution information, and therefore an optimal approximation of the actual (and unknown) distribution $p(\mathbf{\Theta})$.

Within the speech context the dependence tree structure has been used to model the adaptation process and therefore acoustic observations which are used to estimate a single adaptation transform can be spread to other adaptation transforms. [2] [15] [7] This work has shown improvements of up to 4% absolute improvement in WER.

The fundamental algorithms for multi-scale stochastic processes have been developed by Chou, Willsky, and Beneviste, of which [6] is a foundational paper. This paper describes the mutliscale process in a format analogous to the well known state-space model. Equation 3.10 is the multiscale equivalent of the state-space model, where t indicates scale and not time.

$$\begin{aligned} x(t) &= A(t)x(\pi(t)) + B(t)w(t) \\ y(t) &= C(t)x(t) + v(t) \end{aligned} \tag{3.10}$$

In equation 3.10 $y(t)$ represents the observations as generated by the state $x(t)$ and the transform matrix $C(t)$ with additive white noise $v(t)$. The state $x(t)$ is a function of the parent node $x(\pi(t))$ transformed by $A(t)$ with transformed process noise $w(t)$. The approach described in [6] and [8] represents a Kalman filter on a tree, which consists of the standard prediction then filtering steps, as well as a merge step which is not found in typical time-linear dynamic systems. This filtering process takes observations $y(t)$ and spreads them upwards through the tree. This is followed by a smoothing step by the Rauch-Tung-Striebel algorithm which then smoothes the observations back down the tree to leaf nodes. This approach assumes the tree process A, B, C is already defined.

The outstanding problem of estimating the parameters of the tree process A, B, C has been addressed by Ronen *et al* in [19] and [22]. This approach uses the EM algorithm to iteratively estimate the state occupation of acoustic data, then maximize the likelihood of the states given the associated data by adjusting parameters. The algorithm presented is able to make use of the RTS algorithm to smooth model estimates from data at some nodes to other nodes where data is sparse or non-existent. On this topic, an interesting discussion of what types of stochastic models can be estimated by Baum-Welch type EM algorithms can be found in [17].

Chapter 4

Cross Task Speaker Adaptation

While it has been common to perform within task adaptation, cross task adaptation has not been explored. Cross task adaptation makes use of a base model set trained from data accumulated in one environment, and then adapted to a different operating environment. It is frequently the case that the operating conditions of an ASR system will vary greatly from the environment used to train the initial models. The experiments conducted here make use of the large amount of Switchboard data to train initial systems, followed by adaptation to the Macrophone task. The software tool HTK has been used to perform initial model training and to implement MLLR adaptation.

4.1 Description of Data

The 10 hour subset of Switchboard was used for initial model training. This data set has been specifically selected from the entire Switchboard corpus to be representative of the full corpus both in terms of phonetic coverage and speaker variance. The 10 hour subset consists of 22,000 utterances by 2500 speakers, both male and female, recorded over T1 long distance telephone lines and stored in 8-bit mu-law format. A total of 6500 words and 7,000 triphones are represented in this set. The speech segments consist of sentence fragments generated by an automatic segmentation program.

For adaptation and testing, a set of 809 utterances from 41 speakers, all male, were selected from the Macrophone corpus. This corpus contains natural speech read over T1 telephone lines directly to an automated attendant. The data set was split into adaptation and testing, with 48 minutes of adaptation data total (an average of 70 seconds per speaker) and 44 minutes of test data total (an average of 65 seconds per speaker). Macrophone utterances consist of a combination of words, phrases and numbers.

The data has been parameterized into MFCC format with 12 cepstral coefficients and energy, as well as the first and second derivatives (velocity and acceleration) for a total of 39 coefficients per 10ms sample period. The sampling window is 25ms, and a Hamming window with pre-emphasis is used. For acoustic modeling, a 39 phone set taken from TIMIT has been used, with the addition of a long silence model and a short pause model, neither of which are included in context sensitive models. Decision tree clustered 5 mixture, 3 state, left-right Gaussian HMMs with diagonal covariances are used to model the 7200 triphones which occur in the 10K word multiple pronunciation dictionary.

4.2 Experimental Results

This section describes the results of selected experiments in MLLR cross-task adaptation. The software package HTK [3] was used for training base HMMs and for performing MLLR adaptation. Unless otherwise stated for a particular experiment, the configuration parameters were set as described in the following paragraphs.

Adaptation of means and variances were performed using block diagonal adaptation matrices. Three blocks were used to reduce the complexity to 1/3 of a full matrix. This makes the assumption that the static, velocity, and acceleration coefficients are independent of each other. The Word Insertion Penalty (WIP) was set to -10, and the Grammar Scale Factor to +10. These settings achieved a reasonably close %DEL and %INS, both around 2 to 3%. The Baum-Welch re-estimation had a pruning threshold of 350. The regression trees were generated using a binary splitting algorithm based on Euclidean distance, and trees were generated until the specified number of leaf nodes existed. Node thresholds for estimating an adaptation transform was set to 700. The language model was produced from the Macrophone data set, and the dictionary contained 5334 words representing the adaptation data set, with pronunciations coming from the

Description	WER	Expt.
Baseline - no adaptation of 10hr HMMs	24.07%	1
Global transform only	24.07%	2
Unsupervised MLLR, update after every utterance	21.76%	7
Supervised MLLR, 16 leaf regression tree	18.21%	13
Supervised MLLR, 32 leaf regression tree	18.21%	11
Supervised MLLR, 64 leaf regression tree	18.21%	16
Supervised MLLR, 128 leaf regression tree	18.21%	14
Supervised MLLR, RT-64, 39 block transform	22.83%	15
Supervised MLLR, RT-128, generic LM and dictionary	45.55%	9
Supervised MLLR, RT-128, threshold=50, 60-70 adapt classes	57.43%	17
Supervised MLLR, RT-128, threshold=100, 30-40 adapt classes	27.26%	18
Supervised MLLR, RT-128, threshold=200, 12-20 adapt classes	20.23%	19
Supervised MLLR, RT-128, threshold=400, 6-10 adapt classes	18.37%	20

Table 4.1: Summary of Significant Experimental Results

CMU dictionary version 0.6. Adaptation was done in batch on approximately half the data per speaker and used to create a set of adaptation transforms depending on the regression tree node occupancies. Adaptation results were generated by applying the estimated transforms to the remaining half of the data. In all cases both a single global transform and MLLR transforms were applied to the Switchboard models. Details of the implications and selection of these values can be found in Chapter 5.

Table 4.1 lists the notable experimental results. From these, it can be seen that almost all adaptation configurations gave substantial improvements in WER from the baseline 24%. It is common to use all adaptation data and estimate a global transform. In this case, the global transform alone gave no improvement in WER. Analysis of the recognition results and the global transform file reveal that word likelihoods and boundaries do shift, and that the global adaptation does produce a non-trivial transform, however the resulting word sequence is identical, therefore the WER is unchanged. This result also suggests that the LM may have a dominant effect on determining word sequence.

The unsupervised MLLR adaptation results are quite promising, considering no transcriptions

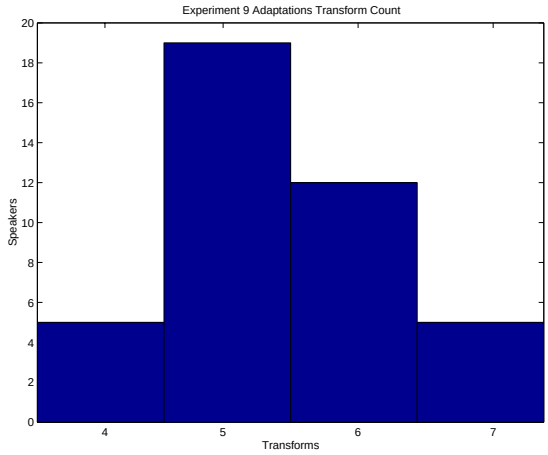

Figure 4.1: Transform distribution for default node occupation threshold (700)

are provided to correctly assign adaptation data to a class. This has to be done based on the recognizer output. While these results are positive, with over a 2% improvement in WER, updating the model after ever utterance is generally not practical since it is computationally expensive. As well, it shows a limit of performance for unsupervised MLLR adaptation, since any batch unsupervised adaptation would be expected to have a higher WER.

The first four supervised MLLR experiments listed in table 4.1 seem to indicate that changing the size of the regression tree did not have any effect on these experiments. In fact, it turns out that the node occupancy threshold of 700 results in only 4 to 7 adaptation classes per speaker given the one minute or so of adaptation data that was available. This can be seen in figure 4.1, where experiment 9 had the identical transform structure to experiments 11,13,14, and 16. In this case every tree structure is forced to regress to a tree of effectively only 7 leaf adaptation nodes, which is smaller than even the 16 node tree. Therefore the 16,32,64, and 128 leaf regression trees all have identical adaptation transforms.

Using a diagonal transform which eliminates cross-parameter correlation clearly is a disadvantage. In the case of experiment 15, using 39 elements for the block transform gave a 22.83% WER. This is equivalent to making the transform matrix diagonal. This result, while still more than 1% better than the baseline, is almost 3% worse than the best case, indicating that cross-parameter correlation should not be ignored and is an important features which can be utilized to improve parameter estimation and adaptation.

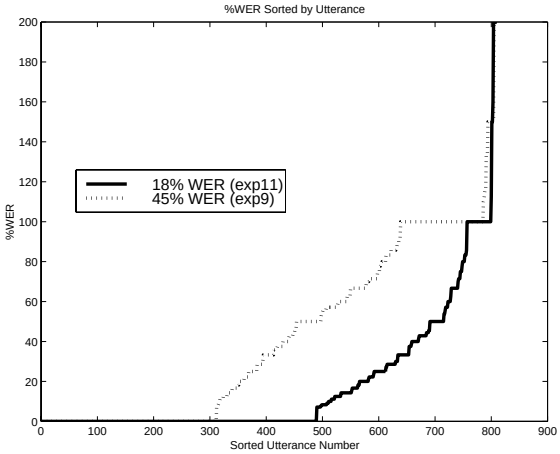

Figure 4.2: WER Comparison for Experiment 9 and Experiment 11

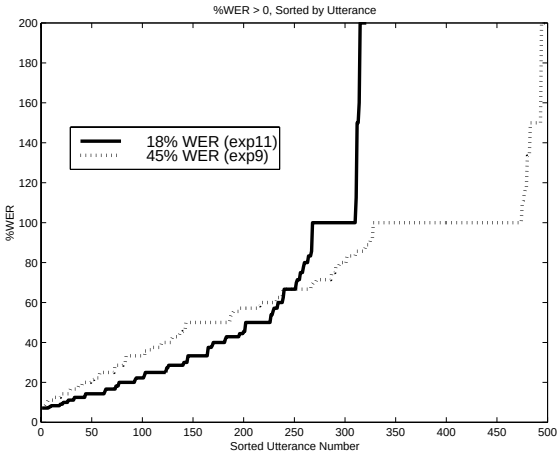

Figure 4.3: Non-Zero WER Comparison for Experiment 9 and Experiment 11

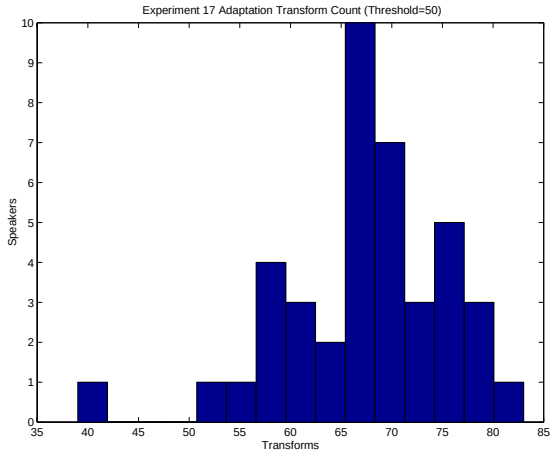

Figure 4.4: Transform distribution for node occupation threshold of 50

To investigate the effect of the LM and dictionary on the result, a generic LM and accompanying dictionary were used. This produced dramatically worse results, with a WER of 45.55%. While in one sense this can be attributed to the importance of a reasonable LM, it is more likely due to the high weighting which the LM was given, since the GSF was set to 10. Such a high value is usually only used when the LM is known to closely represent the testing environment data. In any case, figure 4.2 shows the sorted distribution of the WER by utterance for both experiment 9 (45.55% WER on average) and experiment 11 (18.21% WER on average). Figure 4.3 highlights the fact that the non-zero WER utterances largely have a similar distribution, so the improved model has simply shifted the results along the curve. It is also interesting to note that WER above 100% are relatively infrequent, occurring only 25 times (3.1% of all utterances) in experiment 9 and 11 times (1.3%) in experiment 11, however they contribute a much larger proportion to the overall WER average. If occurances of WER > 100% were reduced to 100%, the effect would be an average WER of 40.7% for experiment 9, a reduction of almost 5%. For experiment 11, since there are fewer occurances of high WER utterances, the reduction would only be 0.6% to 17.6% WER on average.

The final four lines of table 4.1 show the effect of changing the number of adaptation classes. An obvious conclusion is that the node occupancy threshold should be used to set the effective tree size, rather than the original splitting algorithm which creates the binary regression tree. The tree generating process can be done quickly, and therefore some large power of 2 number of

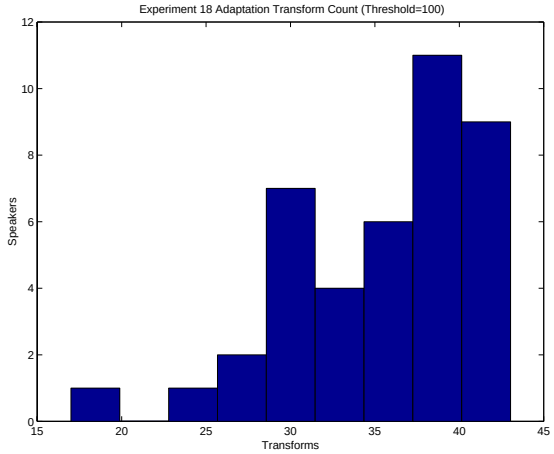

Figure 4.5: Transform distribution for node occupation threshold of 100

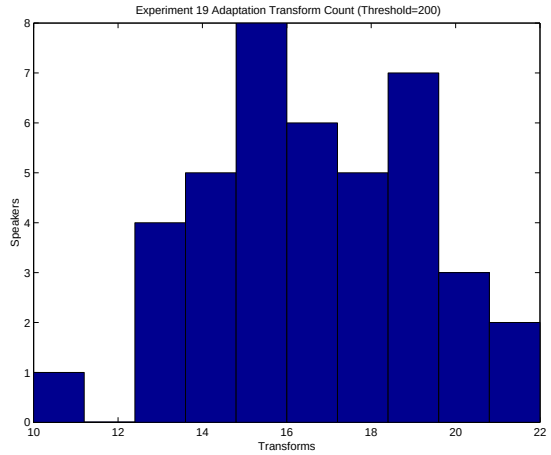

Figure 4.6: Transform distribution for node occupation threshold of 200

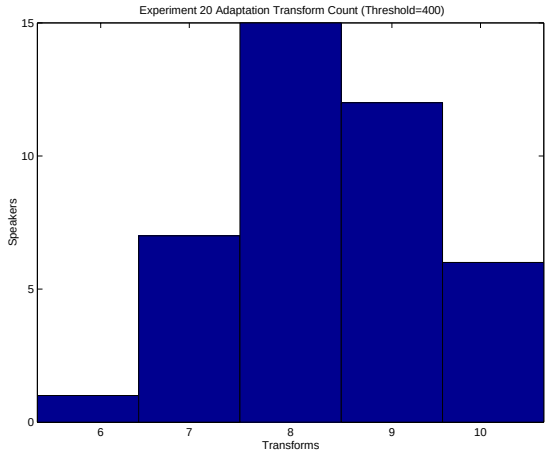

Figure 4.7: Transform distribution for node occupation threshold of 400

leaf nodes should be specified, then the node occupancy threshold can be adjusted to generate an appropriate number of adaptation classes given the available data. Figures 4.4, 4.5, 4.6, and 4.7 show the distribution of transforms to speakers. The original threshold of 700 produced the best results with 18.21% WER, and and 4 to 7 adaptation classes per speaker.

In the extreme, using a threshold of 50 produced 60 to 70 adaptation classes per speaker, which means approximately 1 second of data was available per transform. For a 3 block adaptation transform of means and variances (variances are diagonal and are only transformed by a diagonal matrix), a total of 585 parameters need to be estimated. One second of data contains a total of 100 39 parameter vectors, or 3900 values total. This only amounts to 6.67 samples per parameter, assuming an equal spreading of adaptation data between transform classes, which is probably far from the truth, meaning some transforms would have parameters estimated from perhaps as few as 1 or 2 samples. In this case it is easy to understand how the transforms would be detrimental and not representative of general class transformation. At a threshold of 100 the number of adaptation classes is still 30 to 40, so the problem remains with an average of only 13 samples per estimated parameter, and the resulting WER is 27.26%, which is worse than the baseline result.

Occupancy threshold limits of 200 and 400 produced 12 to 20 and 6 to 10 adaptation transforms respectively. These both improved recognition results over the baseline, but not as well as the 700 threshold system. With so few adaptation classes, it is clear that only the coarsest adaptation features are captured. This illustrates a clear disadvantage of MLLR over correlated adaptation

structures such as Multiscale, in that large amounts of data are required if fine adaptation features are to be captured.

4.3 Summary of Results

MLLR has the potential to dramatically improve recognition rates with relatively small amounts of data. The cross-task experiments discussed above show a 6% (absolute) reduction in WER for 41 speakers after one minute of adaptation data has been collected. The regression tree structure was entirely controlled by the node occupancy threshold, therefore it is reasonable to conclude that a fairly large regression tree should be produced in the first case, probably with 256 or more leaf nodes. A power of 2 number is only relevant when dealing with a binary tree, as it helps to maintain a well balanced tree. The effective regression tree is then established by the node occupancy threshold, using the original regression tree as a base. Cross correlation of parameters is clearly an important feature to make use of in adaptation transforms, as it contributes 4.5% to the WER reduction.

It is also not clear that the regression class clustering approach of using Euclidean distance is optimal, since it can be hypothesized that two acoustically similar class from the original domain may, in fact, require adaptation transforms which are not co-linear. Work by Ostendorf et al [2] has suggested the use of decision trees to generate the tree structure. The decision tree is generated using a combination of heuristic acoustic questions and maximum likelihood. This form of hierarchical correlation structure is already applied for state tying of HMMs, and could be transfered to the tree structure used by the MLLR process.

Chapter 5

Design of HMMs and the ASR Experimental Process

While HMMs are presently considered the state of the art model, there are still extensive decisions that need to be made when creating an HMM set to model human acoustics. Furthermore, the HMM training methodology and ASR experimental structure introduce complexities that can have dramatic effects on system performance, depending on which decisions are made. This chapter discusses the process of creating typical multi-mixture gaussian triphone HMMs for performing ASR experiments. The majority of the material is equally relevant to non-Gaussian and non-phone based models. The general objective task is speaker independent large vocabulary continuous speech recognition (SI-LVCSR). The discussion is focused around the use of the software package HTK by Entropic, as this is viewed as the preeminent HMM research software, and is widely used both in academic and commercial research settings.

In theory, the complete ASR experimentation process requires only the following components:

- waveform speech data
- word level transcriptions for at least 1 hour of waveform speech data
- pronunciation dictionary containing acoustic units of interest (e.g. phone level pronunciations)

- language model
- modeling format

Even the language model is not strictly necessary, since it can be generated either from a word loop using the set of desired recognition words, or statistically generated from the data transcriptions. Preferably this should be done using only the training data transcriptions, otherwise the model has *a priori* information regarding the test data, which will result in an unjustified performance improvement.

These components alone can be manipulated to form all the necessary pieces for a complete ASR system. The following sections describe the specifics of each part of that process in approximately the order in which they are executed when developing an ASR system.

5.1 Data Collection

In practice, much more than 1 hour of training data is required for a robust LVCSR system, although under appropriate task constraints it may be that several minutes of transcribed data is sufficient for training an initial system. While it is desirable to have transcriptions which contain alignment (i.e. timing) data, this is usually impossible and automatic alignment algorithms must be used.

To confirm system performance once a model is fully trained, it is necessary to reserve a portion of the transcribed data for testing. The challenge is that a fixed amount of transcribed data is available, and the quality of training is directly proportional to the amount of transcribed training data available, therefore it is necessary to reserve the smallest amount of transcribed data for testing that will still provide accurate estimates of system performance. In practice, this is a difficult task and usually requires manually selecting representative components of the available data, usually making up 10% to 20% of the total. It is also possible to perform testing recognition on the training data set to indicate the difference in the model fit between the two sets – this shows how representative the test set was of the training set.

It is important to be aware of the variance that exists in recognition results depending on the "luck" of the test set quality. Figures 5.1 and 5.2 show the %WER and %COR for experiment 13

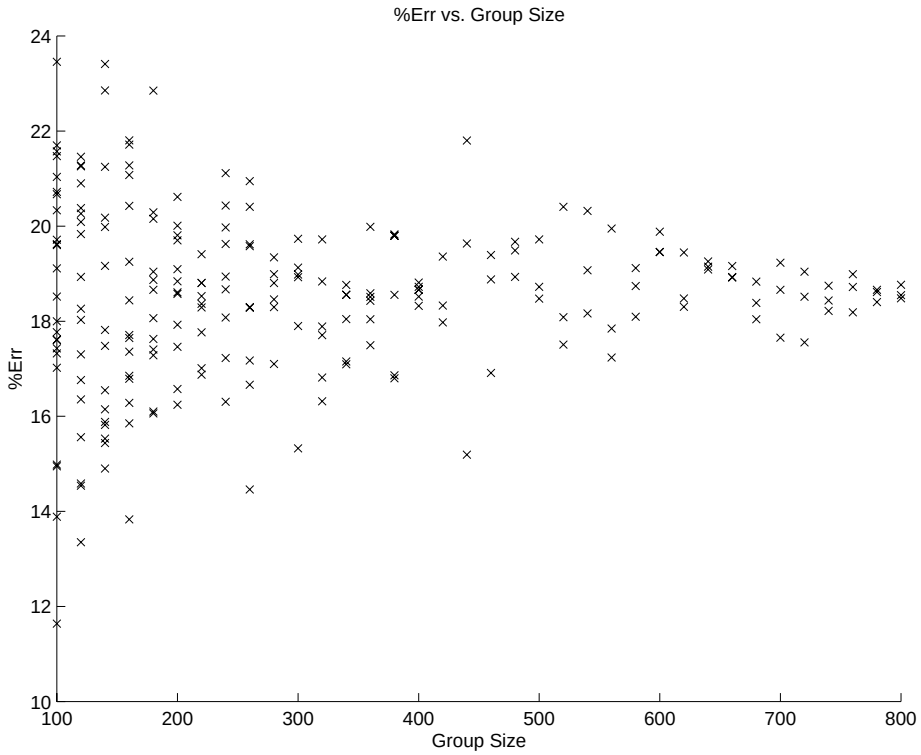

Figure 5.1: %WER for randomly selected groups of test data

(see chapter 4) which had an average %WER of 18.21% for all 809 test utterances and a %COR of 83.9%. It can be seen that at a group size of 440 (which would be over 22 minutes of data), %WER varies between 15% and 22% and %COR varies between 81% and 87%. Furthermore, the convergence of these results to the value of 18.21% WER is only due to the limit of the total amount of testing data being used, so it can be speculated that with more data the "actual" %WER for this task could vary by +/- several percent. It is for this reason that more than one hour data is usually required for the test set in order to make a reasonable estimate of the system performance (i.e. bounded to within +/- 5% relative to the estimated %WER).

In terms of units, it is best to separate from the outset a training set of files and testing set, both in terms of the waveform files, the transcriptions, and the parameterized files. It is general practice to have no speaker overlap between train and test sets, and if multiple speaker styles exist in the overall set, to insure that the test and train sets proportionately represent this variation. Usually, gender and dialect regions are the two main variations, but age and recording quality,

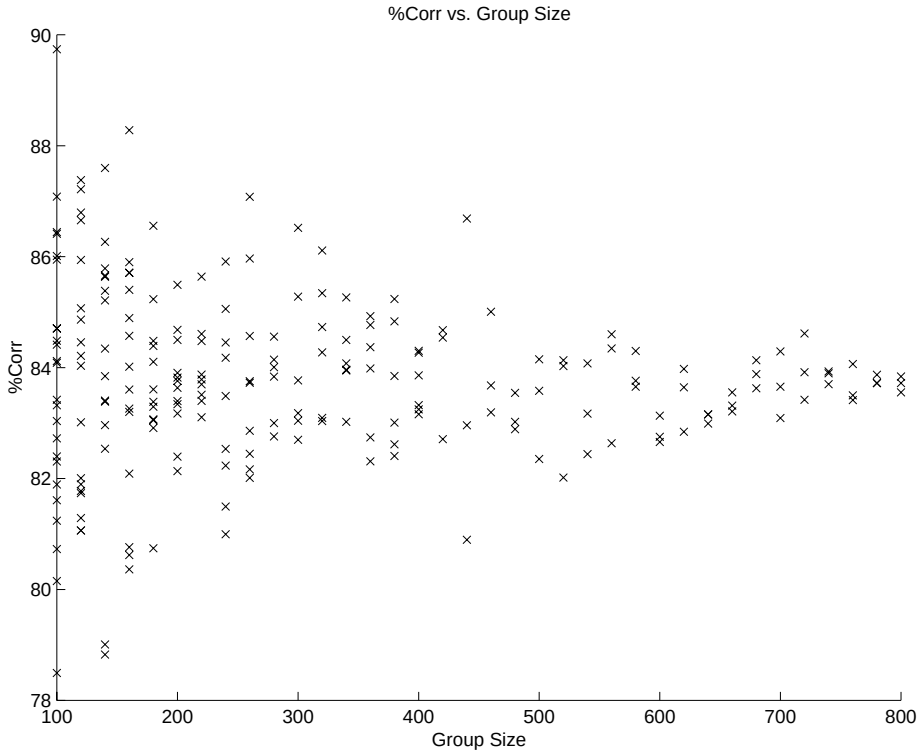

Figure 5.2: %COR for randomly selected groups of test data

for example, may also be factors which require balanced representation.

5.2 Dictionary

The pronunciation dictionary should contain all the words in the available transcriptions and must contain all words in the language model since LM word networks are expanded to phone networks using the pronunciations. CMU has produced an extensive dictionary containing over 120,000 words (counting multiple pronunciations) which is freely available. If the set of missing dictionary words is small, it may be possible to manually create dictionary entries for them. Otherwise, it is necessary to assign some catch-all model. This is often necessary when some general LM is used for which not all LM words are available in the dictionary.

The dictionary contains context free phonetic pronunciations and therefore sets the base monophone set which is used in the model. Using multiple pronunciations provides the opportunity to better match the phonetic transcriptions to the actual data in the training stage, however it increases the LM decoding complexity during the recognition stage and this added confusion may decrease performance by more than the gains made from the benefit of representing multiple pronunciations.

The dictionary is also used to expand the word level transcriptions to phone level transcriptions. This process sets the active phone set for the system. When using monophones, it is likely that the entire monophone set will exist with as little as a few minutes of data, however it is important to insure that this is the case (i.e. the phone /*jh*/ may not occur in the transcription set, but is required since words in the LM have pronunciations requiring /*jh*/). With monophones, it is very difficult to overcome the problem of missing phones, so either additional training data must be collected or a remapping of the phone set occurring in the dictionary must be made in order to insure that all phones occurring in the dictionary are represented by the training data set.

This can be particularly a problem when multiple dictionaries have been merged. The common approach for multiple pronunciations is to place one dictionary entry consistently first followed by the others. When the transcriptions are expanded, the first case of all multiple pronunciations will always have come from the same dictionary. If the "primary" dictionary was exhaustive and listed first, then the phone level transcriptions will *only* contain phones occurring in the "primary"

dictionary. If the alternate dictionaries contain different phone sets, then none of the alternate phones will be represented in the transcriptions and therefore will have no chance of receiving any training data. For this reason it is critical to compile a list of all phones occurring in the phone level transcriptions and all phones occurring in the master dictionary. These lists must be identical, and any phones occurring only in the dictionary must be remapped to phones which are represented in the transcriptions.

The problem of missing phones is almost impossible to avoid when using triphones, since only triphones occurring in the training data can be modeled. If multiple pronunciations are used, or the LM contains words not occurring in the training data set, then it is likely that there will be missing triphones when it comes to testing the ASR system, and the testing will fail. To overcome this, it is necessary to create two triphone lists. One is generated from all triphones occurring in the training data *after* it has gone through pronunciation alignment (see section 5.6.2), which represents the set of triphones which can be modeled by the training data. The second list is generated from a triphone dictionary which is limited only to words occurring in the LM which will be used for testing, and containing all pronunciations for all words (if multiple pronunciations are to be used). If no LM is known at the time, then all triphones in the desired dictionary must be used. With these two lists it is possible to determine the set of triphones which will *not* be modeled by the training data. The only good way to generate these models is to use a decision tree clustering algorithm which will allow unseen triphones to be synthesized (see section 5.7.1). If it is not possible or practical to create a decision tree, then unseen triphones can be copied from the monophone model for the center state, or can be generated using manual tying, although neither approach can be expected to produce desirable results.

5.3 Language Model

There is a tight relationship between the LM, the dictionary, and the transcribed data. Firstly, recognition is strictly limited to words which exist in the LM, and since recognition is generally done on a sub-word unit, the LM is expanded using the dictionary into a sub-word unit network. This implies that all entries in the LM must occur in the dictionary, and no words outside of the LM can be recognized.

To recognize more words than exist in the LM, a simple word loop LM must be created using

the dictionary word list. Then all words in the dictionary occur in the LM and can therefore be recognized. In a word-loop LM every word has an equal likelihood of following the current word. An improved model will take a linear combination of a word loop model and a statistical or heuristic LM. If LM words do not exist in the dictionary, it is possible to create a catch-all model for those words, simply to satisfy the condition that LM entries must have a dictionary pronunciation. This catch-all model should be sufficiently generic as to not ever be mistakenly recognized. It then becomes important to check that the catch-all words do not appear in the recognition output.

The LM can be generated from the transcriptions, however only word sequences which occur in the transcriptions will be valid recognition sequences. For this reason, it is usual practice to generate a LM from some large body of text, preferably spoken. Written syntax has a distinctly different structure than spoken syntax. Furthermore, a "pure" LM should not make use of the testing data transcriptions, however this can make recognition of test data almost impossible for unconstrained word sequences, since any sequence not contained in the training set for the LM has 0% probability of being accepted as a recognition result. Because of this, combining a statistical LM with a word-loop LM allows a non-zero probability of any word sequence occurring. The final option is to use a "cheating" LM which is trained using the testing data LM. If only the test data transcriptions are used, this can be seen as providing a best case limit on the performance of an optimized LM, however it should also be accepted as an unrealistically good result.

5.4 Data Parameterization

Speech data for adults is effectively band limited to 8 KHz for adults, and 10 KHz for young children. This implies a source sampling rate of 16 to 20 KHz, although the application of companding algorithms such as μ-law or a-law are typically used, and corpus' which use speech recorded over telephone lines are limited to 4 KHz acoustic bandwidth or 8 KHz sampling rate. Obviously, this has important implications for the representation of HF acoustic components, which are typical for fricatives and stops.

Source data should generally be recorded at 16 KHz or above in order to capture the full speech spectrum – it should be noted that bandwidth of the human voice and auditory system is, in fact, around 20 KHz, but features above 10 KHz typically only occur during singing and have

not been shown to greatly influence speech information.

At this stage it is necessary to determine a parameterization format, parameter sampling rate, and any windowing or pre-emphasis. Typically, 10ms sampling rate with a 25ms window using a 12 MFCC parameterization is chosen, with added energy, velocity, and acceleration for $(12 + 1) \times 3 = 39$ parameters per 10ms sample vector. This is equivalent to 3900 parameters per second being generated from the original 8000.

The raw acoustic longitudinal wave amplitude is not a useful representation of the acoustic features, therefore it is standard practice to parameterize the acoustic sequence into a more structured format. Estimating the linear prediction coefficients of the all-pole model has been popular[13], however MFCC is preferred. This also necessitates specifying a vector sampling rate, for which 10ms has been shown to be suitable for capturing short and long acoustic phenomena. MFCC makes use of a discrete Fourier transform (DFT) to generate frequency components. To avoid discontinuities at the DFT window edge, an overlapping window of 25ms is typically used along with a Hamming window which attenuates the signal to zero at the edges, thereby eliminating the HF noise incurred by a discontinuity. The N MFCC coefficients are generated from an inverse DFT of a mel scaled filter bank. A pre-emphasis coefficient of 0.97 is also standard for boosting the HF signal since perceptual loudness is different from actual signal strength.

5.5 HMM Template

The HMM is used to model the acoustic units. While it is theoretically possible to have different HMM structures for different acoustic units, in practice all HMMs are the same thereby producing a uniform model set. The most general HMM consists of state models and a transition probability matrix. Non-emitting entry and exit states are used to facilitate merging multiple models into one meta-HMM. Non-emitting states do not produce observation vectors and immediately transit to the next state.

HMMs generally are static stochastic models of the observation vectors produced by a particular acoustic unit. They are static in the sense that while in a particular state the distribution of the observation vector is fixed. This limitation is not intrinsic to the HMM, and work has been done to use trajectory models where the output distribution follows some path rather than a fixed value.[15] [28] Figure 5.3 shows the differences between static, linear, and quadratic models, with

different numbers of states to make the total number of model parameters equivalent. There are significant complexities involved in using non-static state models, however the benefits to better matching the actual parameter trajectories may prove to be worth the added computational load. The trajectories in figure 5.3 are unconstrained, meaning there is no continuity constraint, however continuity can be asserted, but again at a significant computational cost, since this requires transferring information about the terminating point of the previous state, and optimizing that point.

Three states are usually considered sufficient when models are on the phone level. There is also the attraction that the boundary states can be associated with the context models, for example the first state being associated with the coarticulation features of the left (previous) phone. The models for each state are typically represented by Gaussian distributions of the observation vector. In most cases, independence of the individual parameters are assumed, which results in a diagonal covariance matrix, thereby greatly simplifying parameter estimation. It should be noted that this simplifying assumption is known to be poor since parameters are, in fact, closely correlated. By the independence assumption each state has $2\dot{N}$ parameters, where N is the size of the observation vector (each coefficient has a scalar mean and variance). Even using block diagonals to assert independence between static, energy, velocity and acceleration coefficients still results in $3\dot{N}^2 + N$ parameters. For 12 static coefficients the difference is 78 parameters assuming independence or 474 parameters using correlation by block diagonals. A full correlation matrix would have 1560 parameters. Finally, many algorithms have been developed which only work on diagonal covariance matrices. This is frequently the case in HTK.

Given the sequential time structure of the physical phenomena being modeled by the HMM for speech, it is reasonable to assert a left-right state sequence, meaning that states must occur in order and that it is never possible to return to an earlier state. State transition likelihoods can be estimated from training data so initial values should simply contain approximate values for initialization, such as a 70% likelihood of remaining in the current state and a 30% likelihood of transiting to the next state. Figure 5.4 shows the structure of a typical Gaussian HMM.

It is necessary to have one model per acoustic unit, and typical tri-phone training consists of an iterative process which begins with only monophone models. In this case, the set of 40 to 50 monophone models must be initialized to reasonable values to begin the training process. Since pre-established initializations for each model usually aren't available, a global model is generated

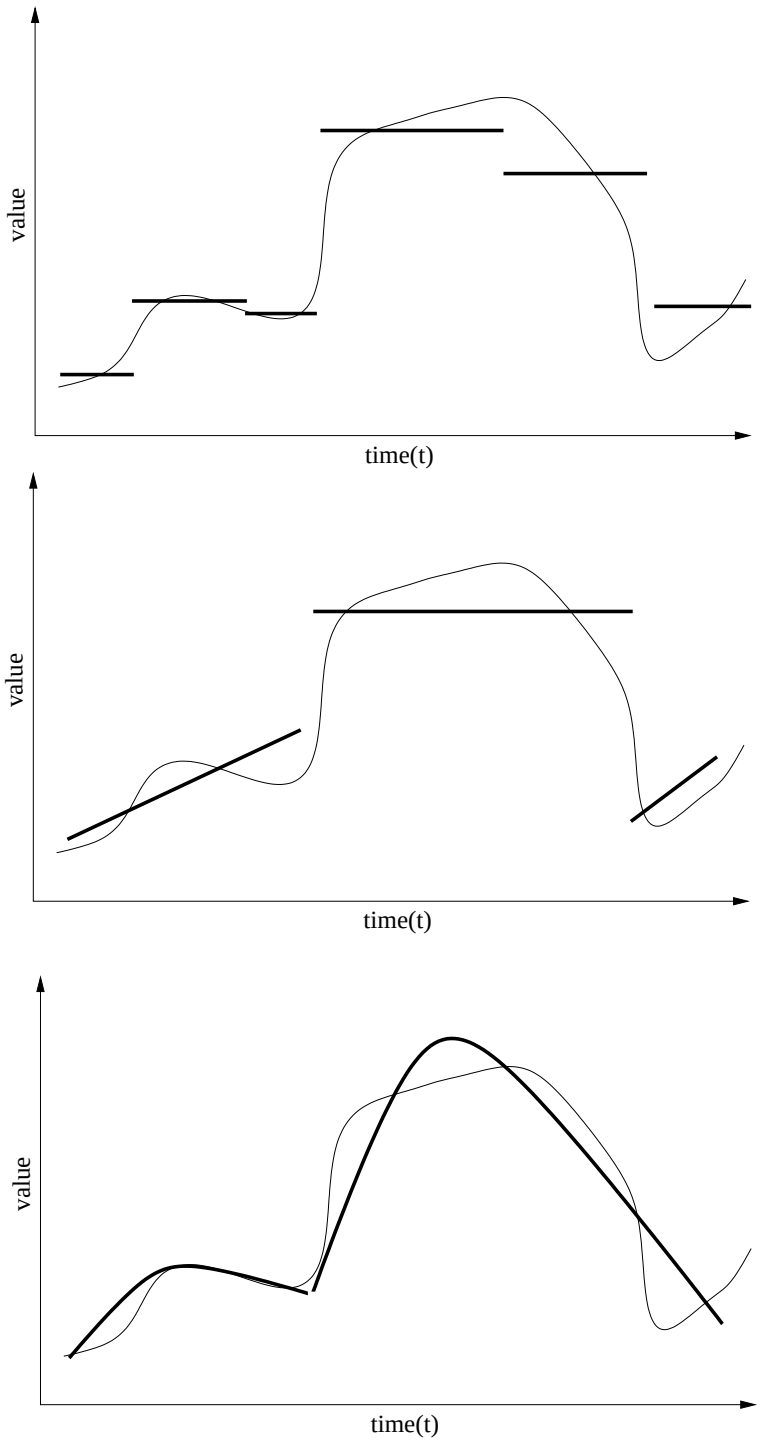

Figure 5.3: Comparison of different order models

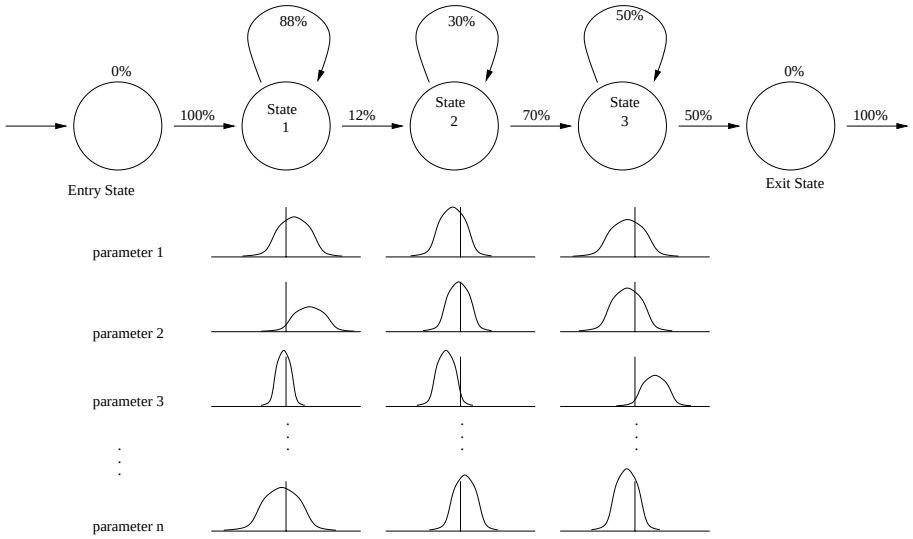

Figure 5.4: Structure of a left-right 3 emitting state Gaussian HMM

by accumulating statistics for the entire training data set. This global model is then copied to ever monophone model as a starting point.

At this stage a set of identical HMMs, one for each monophone, will exist. The models will contain statistical distributions for each state within the model, and a state transition matrix.

5.6 Training Monophone Models

Since alignment data is typically not available with transcriptions, it is necessary for the Baum-Welch algorithm to best estimate the model and state transition points. Given that these estimates are made from the existing models, it is plain to see that initial boundary estimates will be poor. It is necessary to use as much data as possible to improve the monophone models discrimination ability before fragmenting the model space significantly through the use of triphone models. Furthermore, the models are very sensitive to noise and poor training data since every model is initially identical. For this reason, it is beneficial to use a boot strap portion of the training set which has been checked to confirm that the transcriptions are accurate and the data is relatively noise free and of clear audible quality.

The basis of the HMM training process is the Baum-Welch algorithm which uses current data

to estimate transitions and state sequence likelihood, then optimizes the models to best match the assigned state sequence. This is a form of the EM algorithm and is intrinsically iterative in nature, slowly converging on the (locally) optimal model, since the EM algorithm is proven to not degrade the models with successive iterations. After 2 or 3 iterations of retraining with the bootstrap portion of the training data, the full training data set may be used (including the SP and SIL models).

During these training iterations there are only three user decisions: i) the number of iterations to use; ii) the pruning threshold; and iii) variance floor. It has been found in general that the Baum-Welch algorithm performs the majority of it's improvements in the first 2 or 3 iterations, therefore no more than 3 iterations should be made after a change in the model. The pruning threshold is related to the automatic alignment of transcriptions with data. Since numerous segmentation intervals are possible, the search algorithm can be limited by pruning combinations which fall a certain distance below the maximum likelihood. For monophones, with only 40 to 50 models total, pruning is not necessary. The variance floor prevents data sparsity from producing a model with an extremely small variance. By setting a variance floor, no variance parameter will be permitted to fall below the floor value.

5.6.1 Modeling Silence

Two additional model types which have not yet been mentioned are the silence model (SIL) and the short pause model (SP). When there are extended periods of silence in an utterance, it is necessary to mark this with an appropriate label, otherwise the training process will be forced to stretch phonetic models to cover the silence region. The short pause model is used to model the word terminus period which may or may not contain silence. In natural speech there is no discontinuity between words unless a pause or end of utterance is reached. The short pause model is an alternative to cross word triphones, since the cross word triphone is difficult to represent in dictionaries and dramatically increases the number of context models. It is generally tied to the silence model and then attached as the last phone in every dictionary word.

The danger of using SIL or SP during the initial training period is that due to poor model discrimination it is necessary for training models to occur with approximately equal frequency. If a few models occur much more frequently than the rest they will become well trained and

therefore match increasingly large portions of the training data, beyond their proper scope. Since SP typically occurs at the end of every word, and most words contain less than 10 phones, SP will have 4 to 10 times the frequency and therefore will become overtrained. In order to avoid this, it is necessary to use a transcription and dictionary set which do not include any SP or SIL models for the first several training iterations.

5.6.2 Use of Multiple Pronunciations

As has already been mentioned, use of multiple pronunciations can have the detrimental effect of increasing the complexity of the LM network and therefore increase the likelihood of confusion, which increases the error rate. That being said, training of models is done from phone level transcriptions which are usually expanded from word level transcriptions (since transcription is generally done at the word, rather than phone, level), so the selection of which pronunciation to use for a given word will have an effect on which phone models are trained. By default word level to phone level transcription conversion programs will use the first occurance of a word, but once the phone set has undergone initial training it is possible to compare the quality of various pronunciations and select the best match.

Even in the case where final recognition is done using a single pronunciation dictionary, it is advisable to perform transcription alignment and pronunciation selection from a multiple pronunciation dictionary. This is generally done after 2 or 3 iterations of full training with the the SIL and SP models included. This will improve the model training and decrease the effects of bad alignment which occurs when a model is associated with data to which it is unrelated, since this degrades the model estimates.

As a final note, changing the phone level transcriptions will effect the triphone set which exists in the transcriptions. Because phone sequences can change, phone contexts can change. Triphone transcriptions and the triphone list are generally only created after alignment has been done.

5.7 Training Triphone Models

After several iterations of monophone training, with the newly aligned transcriptions, triphone models can be created. It is typical to have 6000-8000 triphones created from the base 40 to

50 monophones. Initially, every one of the triphones will be an exact copy of one of the most recently trained monophone models, based on the center phone of the triphone model (i.e. all phones matching $* - p + *$ will be copies of monophone model p). With 3 states per model, this represents on the order of 20,000 states.

5.7.1 Decision Tree State Tying

A few iterations of training with the full state count is necessary in order to establish some model discrimination (i.e. to vary model $l_1 - p + r_1$ from model $l_2 - p + r_2$). After this, it is beneficial to reduce the number of parameters by tying similar states to improve estimation and overall robustness. A variety of methods are available for tying states so that two models share the same meta-state. An obvious approach is to use a Euclidean distance measure between states, and merging closest states until either a maximum cluster distance was reached or until a set number of clusters had been created. While this approach definitely has merit, it can result in loss of fine discrimination between similar states which need to remain distinct.

Decision tree tying builds a binary tree which contains a combination of heuristic and statistical structure. It is created by asking "yes/no" questions regarding states clustered at the root node, and splitting the data to one or the other child depending on the answer (e.g. states answering "yes" to the question go to the left child and those answering "no" go to the right child). The questions refer to the phone model structure from which the state came from, for example "Left Phone is a Fricative" or "Right Phone is /*aa*/". At each level the question which produces the maximum likelihood split is used then that question is removed from the list (since it has already been applied to the data). This process is continued until the questions have been exhausted or the increase in log likelihood falls below a certain value. The root nodes of the trees generally have all states for a given base phone and state position (e.g. all first states of triphone models for which the base phone is /*ey*/). Leaf nodes can then be clustered if they are very close by a Euclidean distance measure.

One of the greatest advantages of this approach is that unseen triphones can be synthesized by traversing the decision tree until a leaf node is reached. The set of decision tree questions should reflect some sort meaningful phone structure or groupings. Although application of multiple mixture decision trees is possible, many algorithms currently available are based on single mixture

Gaussian distributions. Data driven clustering can generally accept any distribution format, while decision trees are currently limited to Gaussian distributions due to the use of maximum likelihood estimates derived from the model rather than the data. Decision trees are currently the most popular form of triphone clustering and state tying.

5.7.2 Estimating General Statistical Distributions

A single Gaussian distribution for a model is generally not reasonable, given speaker variations which are typically clustered such as dialect and gender. For this reason, it is desirable to have more of a general statistical distribution. A general distribution can be approximated by a multiple mixture Gaussian model, and creation of multiple mixture models should be done after state tying has been completed (and a few iterations of re-estimation).

This process is straight forward and consists of using an iterative mixture splitting algorithm where the initial Gaussian is split into two by offsetting the new means by some fixed amount to either side of the previous mean, then re-estimating the distributions. This can be repeated an arbitrary number of times, although re-estimation should be done after every mixture increase, and mixtures should only be increased by one between re-estimation (i.e. single mixture to double mixture, then retrain, double to triple, then retrain, etc.). Of course, M mixtures increases the total number of parameters by M, thereby creating data sparsity problems and poor parameter estimates.

Most practical LVCSR systems find 3 to 7 mixture distributions is sufficient to capture the parameter distribution. Because the total number of model set parameters increases so substantially during this process, and due to the iterative nature of training multi-mixture models, it is usually necessary to apply limits on the computations to reduce the processing time. This is generally done in the form of a pruning threshold which is monitored during the Baum-Welch re-estimation. The Baum-Welch algorithm must check a large number of possible state sequences and state alignments (i.e. actual time boundaries for the start and finish of each state). During this process a beam search will keep track of the most likely state sequence and any sequence which is more than a certain amount below this likelihood is considered to be sufficiently unlikely as to warrant elimination from the list of possible state sequences. It is possible that this process will eliminate all sequences and the utterance will have no acceptable predicted state sequence,

which will result in training failure. To avoid this, the threshold must either be adaptable (i.e. increase if a failure occurs) or set sufficiently large that this will not occur.

5.8 Performing Recognition and Evaluation System Performance

Once a fully trained multiple mixture triphone HMM set has been produced, it can be used to recognize the test set. As has already been noted, it is necessary to have a LM and a corresponding dictionary which contains pronunciations for all words occurring in the LM. The HMM set must contain every triphone occurring in the dictionary, otherwise recognition will fail when the LM is expanded into a triphone network, using the dictionary, and corresponding HMMs cannot be found. The recognition process consists of two parts: i) traversing the phone level lattice representing the LM to generate an acoustic score; and ii) incorporating a LM score relating to the syntactical structure of the utterance.

The lattice traversal is done using the Viterbi algorithm, and is usually computationally intensive, even given the overall efficiency which is gained by using Viterbi rather than brute force calculation. To limit the computational load, a beam search method analogous to the one described in section 5.7.2 can be used. The lattice is traversed using tokens which move through the lattice in time. Each token contains the current lattice path and it's probability to that point. If any token falls a certain threshold below the maximum token at that time it is eliminated.

Words with more phones will have a lower probability because each time a token is passed from one phone to the next the probabilities are multiplied (and all probabilities are less than 1). This means there is an intrinsic bias towards fewer words in the recognizer output. To compensate for this, a word insertion penalty (WIP) is applied. This value is added to every token when it reaches the end of a word. It can be used to encourage more words to be output or discourage the number of words. Furthermore, the scale of the LM score and the AM score will not correspond so it is necessary to scale the LM score to be appropriately sized compared to the AM score. This is usually termed the grammar scale factor (GSF). It is also an important parameter depending on the quality of the LM and the desired effect of the LM on the recognition output. If pure acoustic results are desired, it is possible to use the LM only for expanding the allowed word and

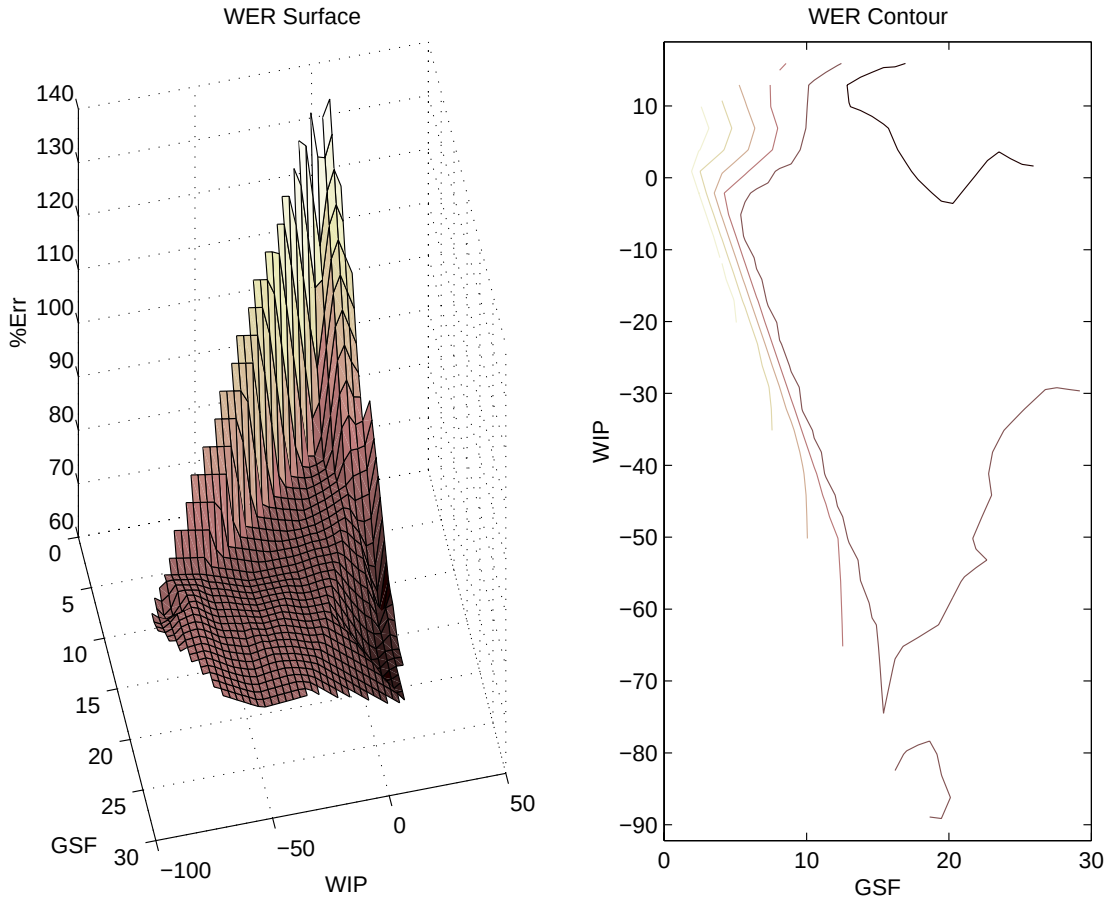

Figure 5.5: WER Surface for various GSF and WIP values

phone sequences, but have no influence on adjusting the token scores by setting the GSF to zero. Of course, even with a GSF of zero, it is possible that the lattice structure of the LM asserts a very high degree of syntactical structure to the recognizer. If pure AM results are desired, it is necessary to produce a word-loop LM using the dictionary entries.

Figures 5.5 and 5.6 show two different WER surfaces generated using various GSF and WIP values. The problem of determining "good" values of GSF and WIP is very difficult. This is usually done from intuition and experience. WIP should be set such that the %INS (word insertions) and %DEL (word deletions) are approximately equal. A high %INS indicates that the WIP is too positive and too many words are being inserted. Instead it should be reduced (made more negative). The converse is true of a high %DEL, and WIP should be increased. WIP can

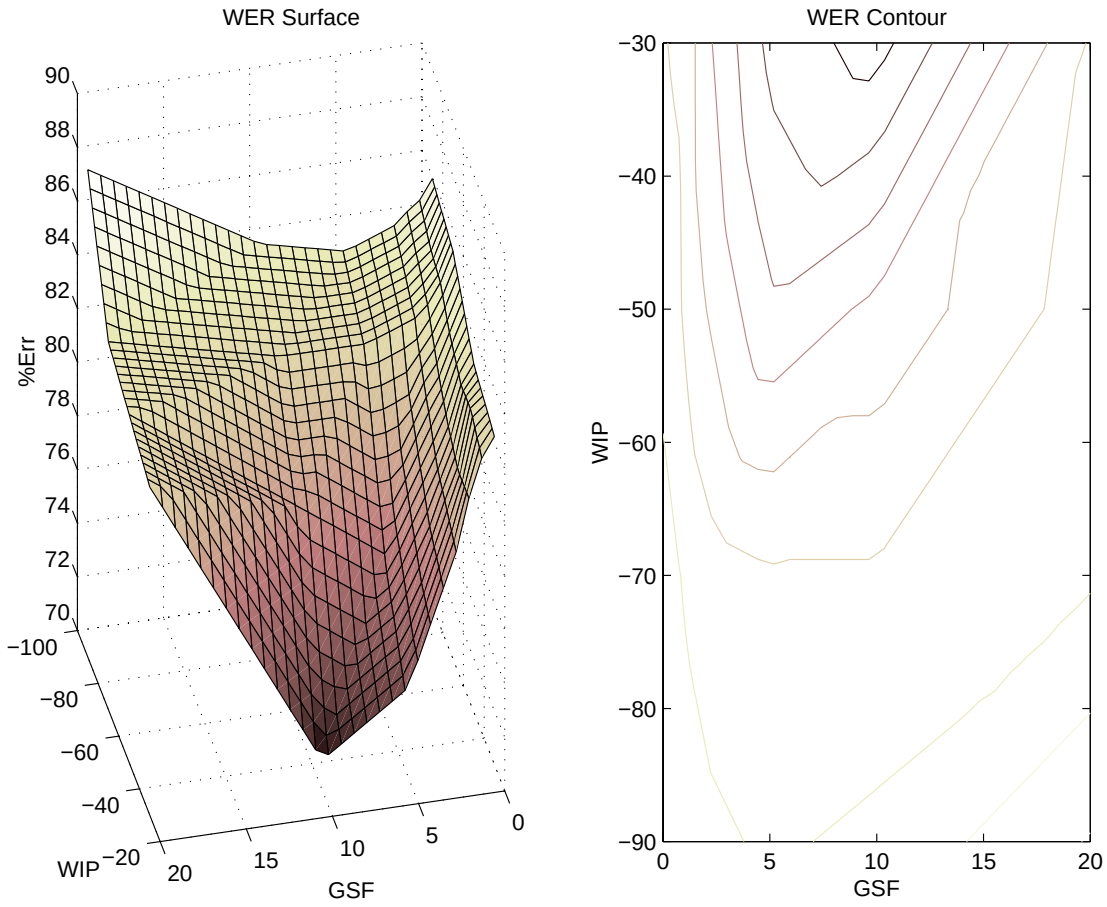

Figure 5.6: WER Surface for various GSF and WIP values

be a positive or negative value, and when it is used as an additive factor for log likelihoods can be expected to be in the range of -50 to +50.

GSF is much harder to establish and is based on the quality of the LM and the regularity of the operational environment (i.e. if the LM is highly structured and the operational environment will have a fairly fixed syntactic structure). The GSF is a strictly non-negative real number, and practice shows that a value on the order of the number of base coefficients scales the LM appropriately to match the AM scores. For example, a system with 12 static MFCC could use a GSF of 12. This is due to the fact that the summation of the token log-likelihoods is dominated by the static coefficients.

In cases where the training data and LM consist of specific sentences it is inaccurate to strongly weight the LM with a high GSF when the test data consists of the same sentences, unless those sentences represent the entire objective operational environment of the system. This problem can best be illustrated by an example. Consider two sentences "Today is Tuesday March fourteenth" and "What would you like to eat?". If a large number of speakers are recorded speaking both sentences and the objective is to train a LVCSR system from these two short sentences, then the testing results, even if "unheard" test speakers are used, will be extremely good if the LM is built from the two sentences. A strong recognition result for any one word in either sentence will automatically fix the recognition result on that sentence. As an extreme case, if a test sentence was completely corrupted by noise except for the word "is", then the sentence "Today is Tuesday March fourteenth" would likely be recognized. It is for this reason that it is very important to have a representative LM that does not artificially have a perfect match to the testing data.

Since the recognition process simply outputs a word sequence, and actual performance measures are done separately by comparing the output to the actual transcriptions, it is possible to do post-processing on the recognition results. Recognition of non-speech events (NSE) may be done by the system, however the actual transcriptions do not consistently list NSEs, so it may be desirable to remove these markers from the recognition results. An important NSE is the $[SILENCE]$ event (or equivalent) which is typically placed at the start and end of all utterances to avoid associating silence with the first and last phones of the utterance. This can easily lead to the recognizer producing $[SILENCE]$ at the start and end of every utterance (especially if it is built in to the LM), which will exactly match the actual transcriptions, and thereby guarantee two correct "words" recognized for every utterance. This will decrease WER substantially, and

with short utterances can lead to highly misleading results (e.g. a set of single word utterances all incorrectly recognized could report only a 33% WER rather than 100%). For this reason it is important to consider what is being produced by the recognizer and what is being scored by the performance analysis software.

The conclusions from these observations suggest that when recognition is performed, the GSF should be set to the base number of data coefficients, and the WIP should be set to balance %DEL and %INS. Pruning should be performed at a level similar to that used in the training (Baum-Welch re-estimation) phase, and the GSF should be adjusted to take into account the quality of the LM. As has been illustrated in figures 5.5 and 5.6, optimal selection of GSF and WIP can improve the WER by several percent, so it can be beneficial to vary these two values and repeat testing to note the effect. Section 5.1 and figures 5.1 and 5.2 discuss and illustrate the importance of having sufficient test data to generate a meaningful measure of system performance.

5.8.1 N-Best Lists and Lattice Results

As a postscript to the topic of evaluating system performance, it is sometimes valuable to be able to analyze the recognition process at a finer level. The Viterbi decoding selects the best word sequence based on a combination of the AM and LM scores (as weighted by the GSF and WIP) given by the equation:

$$\hat{(}w) = arg_w max(L(\sum_w AM + (GSF \times LM + WIP)|w) \tag{5.1}$$

While the normal results are only the highest likelihood sequence, it is possible to have most decoders to output an N-Best list, which contains the alternate word sequences which were not selected. This can be useful to see if the correct word sequence was "close" to being selected. Some ASR systems do sequential improvements by re-scoring from the N-Best list rather than re-computing lattice likelihoods. A further scale of information can be gained by outputting the N-Best lattice structure, which contains lattice connections, alignment, and likelihood information. More advanced refinements can be done with this information than with an N-Best list alone.

Chapter 6

Conclusions

This thesis discusses the development of SI-LVCSR systems using HMMs and for optimisations to improve recognition results. Specific attention is given to the problem of speaker adaptation, and experimental results for cross-task adaptation using MLLR are presented. The experimental process in which HMMs are created and trained is reviewed and analysed to establish important criteria for optimum model design and system performance.

6.1 Adaptation

The task of supervised rapid adaptation from one minute of data was considered. The source models were generated from a 10 hour subset of the Switchboard corpus which contains continuous conversational speech. The target task was recognition of the Macrophone corpus, using 1 minute of supervised adaptation improvements to recognize 1 minute of test data on a speaker by speaker basis. A set of 41 male speakers was used. Using the unadapted Switchboard HMM set a WER of 24.07% was achieved. The best adaptation results were achieved by performing mean and variance adaptation using diagonal variances and performing a global mean transform. This resulted in a WER of 18.21%, or a reduction of 5.86% absolute. Block diagonal transforms were used, and a total of 4 to 7 adaptation classes were estimated per speaker.

These experiments revealed the coarseness of MLLR adaptation and suggested the need to capture distinct adaptation features for acoustically similar models. The adaptation classes were

shown to be established primarily by the clustering threshold, rather than the regression tree size, thus suggesting that a large regression tree should be created and the effective tree size set through the threshold. The cross-parameter correlation was also clearly revealed by the decrease in performance when parameters were independently adapted. In this case the WER increased by 4.62% to 22.83%. This has important implications concerning the possibilities for whitening the parameter space or reducing the number of parameters to increase computation speed.

6.2 Experimental Design

The discussion of experimental design addressed several issues which to date have not been formally covered in the available literature. A presentation on the convergence and variation of recognition results based on test set size suggests that at least one hour of data is required for testing LVCSR systems, and ideally a secondary test set should be available to confirm results. It was estimated that at one hour of test data the estimate accurancy variance will still be +/- 5% of the estimated error rate.

Analysis of the effects of varying WIP and GSF also showed the importance of accurately estimating these values. Error surfaces revealed several percent variation, but also indicated smooth surfaces which ease the task of estimating the optimum settings.

6.3 Future Work

Adaptation is an important part of the progression towards improved ASR systems, especially for speaker independent continuous conversational speech. The work described here has only experimentally analysed one approach to adaptation, while there are several other options such as MAP, EMAP, and multi-scale. It would seem that the current MLLR strucutre suffers from the requiring at least one minute of data in order to perform coarse adaptations, and the ideal system will be capable of performing fine adaptation with only a few seconds of data. Further investigation into the use of correlated adaptation structures such as multiscale should be pursued.

There are clearly still many outstanding issues in the process of optimal experimental design and HMM training methodology. While trigram langauge models have shown significant promise

over bigram models, there is a need to succsefully incorporate the rich information contained in the syntactic and semantic structure of speech. Clearly a recurring theme is the need to overcome data sparsity problems while still retaining discrimination between similar acoustic events.

Bibliography

[1] *Digital Processing of Speech Signals*. Prentice Hall, 1978.

[2] *Computational Models of Speech Pattern Processing*, chapter Tree-based Dependence Models for Speech Recognition. Springer-Verlag, 1998.

[3] *The HTK Book version 2.2*. Entropic, 1998.

[4] J.K. Baker. The dragon system – an overview. *IEEE Transactions of Acoustical Speech Signal Processing*, 1975.

[5] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 1972.

[6] K.C. Chou A.S. Willsky A. Benveniste. Multiscale recursive estimation, data fusion, and regularization. *IEEE Transactions on Automatic Control*, 1994.

[7] V. Digalakis et al. Rapid speech recognizer adaptation to new speakers. Technical report, Johns Hopkins University, CLSP Workshop 1998, 1998.

[8] P. Fieguth. *Application of Multiscale Estimation to Large Scale Multidimensional Imaging and Remote Sensing Problems*. PhD thesis, Massachusetts Institute of Technology, 1995.

[9] J.R. Flanagan. *Speech Analysis, Synthesis and Perception*, chapter 3. Springer-Verlag, 1972.

[10] M.J.F. Gales. The generation and use of regression class trees for mllr adaptation. Technical report, Cambridge University Engineering Department, 1996.

[11] M.J.F. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. Technical report, Cambridge University Engineering Department, 1997.

[12] R.M. Gray. Vector quantisation. *IEEE ASSP Magazine*, 1984.

[13] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America*, 1990.

[14] F. Jelinek. Design of a linguistic statistical decoder for the recognition of continuous speech. *IEEE Transactions on Information Theory*, 1975.

[15] A. Kannan. *Adaptation of Spectral Trajectory Models For Large Vocabulary Continuous Speech Recognition.* PhD thesis, Boston University, 1997.

[16] C.K. Chow C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, 1968.

[17] H. Lucke. Which stochastic models allow baum-welch training. *IEEE Transactions on Signal Processing*, 1996.

[18] L. Deng J.W. Mark. Parameter estimation of markov modulated poisson processes as a telecommunication traffic model via the em algorithm with time discretization. *Telecommunication Systems*, 1993.

[19] O. Ronen J.R. Rohlicek M. Ostendorf. Parameter estimation of dependence tree models using the em algorithm. *IEEE Signal Processing Letters*, 1998.

[20] L.E. Baum T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics*, 1966.

[21] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 1989.

[22] O. Ronen. *Dependence Tree Models of Intra-Utterance Phone Dependence.* PhD thesis, Boston University, 1997.

[23] A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 1967.

[24] C.J. Leggetter P.C. Woodland. Flexible speaker adaptation using maximum likelihood linear regression. In *Proceedings of 1995 ARPAa Spoken Language Technology Workshop.*

[25] C.J. Leggetter P.C. Woodland. Speaker adaptation of continuous density hmms using multivariate linear regression. In *Proceedings of the International Conference on Speech and Language Processing, 1994.*

[26] C.J. Leggetter P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hmms. *Computer Speech and Language*, 1995.

[27] S.E. Johnson P.C. Woodland. Speaker clustering using direct maximisation of the mllr-adapted likelihood. In *Proceedings of the International Conference on Speech and Language Processing.*

[28] L. Deng M. Aksmanovic X. Sun C. Wu. Speech recognition using hidden markov models with polynomial regression functions as nonstationary states. *IEEE Transactions on Speech and Audio Processing*, 1994.

[29] S. Young. Large vocabulary continuous speech recognition: A review. *IEEE Signal Processing Magazine*, 1996.