EFFECTS OF TRANSCRIPTION ERRORS ON SUPERVISED LEARNING IN SPEECH RECOGNITION

By

Ramasubramanian Sundaram

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

August 2003

EFFECTS OF TRANSCRIPTION ERRORS ON SUPERVISED LEARNING IN SPEECH RECOGNITION

By

Ramasubramanian Sundaram

Approved:

Joseph Picone
Professor of Electrical and Computer Engineering
(Director of Thesis)

Nicholas Younan
Professor of Electrical and Computer Engineering
(Committee Member)

Georgios Y. Lazarou
Assistant Professor of Electrical and Computer Engineering
(Committee Member)

Nicholas Younan
Graduate Coordinator of Electrical Engineering in the Department of Electrical and Computer Engineering

---

A. Wayne Bennett
Dean of the College of Engineering

Name: Ramasubramanian Sundaram

Date of Degree: August 2, 2003

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Joseph Picone

Title of Study: Effects of Transcription Errors on Supervised Learning in Speech Recognition

Pages in Study: 52

Candidate for Degree of Master of Science

Supervised learning using Hidden Markov Models has been used to train acoustic models for automatic speech recognition for several years. Typically clean transcriptions form the basis for this training regimen. However, results have shown that using sources of readily available transcriptions, which can be erroneous at times (e.g., closed captions) do not degrade the performance significantly. This work analyzes the effects of mislabeled data on recognition accuracy. For this purpose, the training is performed using manually corrupted training data and the results are observed on three different databases: TIDigits, Alphadigits and SwitchBoard. For Alphadigits, with 16% of data mislabeled, the performance of the system degrades by 12% relative to the baseline results. For a complex task like SWITCHBOARD, at 16% mislabeled training data, the performance of the system degrades by 8.5% relative to the baseline results. The training process is more robust to mislabeled data because the Gaussian mixtures that are used to model the underlying distribution tend to cluster around the majority of the correct data. The outliers (incorrect data) do not contribute significantly to the reestimation process.

DEDICATION

Dedicated to my parents and sisters,

for their constant support and innumerable sacrifices.

ACKNOWLEDGMENTS

First and foremost, I would like to express my gratitude to my major advisor, Dr. Joseph Picone, for his constant inspiration, expert guidance and constructive criticism throughout this work. I have been fortunate to work as a graduate assistant under his guidance. I did learn a lot from his constant mentoring and his passion for perfection. He has been a tremendous influence in my life.

I would like to thank Aravind Ganapathiraju for his valuable suggestions throughout this work. He has been a constant source of inspiration and encouragement. He has been my first critic for this thesis. His patience in reviewing this work and suggesting improvements is greatly appreciated.

I am deeply indebted to Jon Hamaker for his constant support throughout this work. It has been a wonderful experience working with him. His contribution to the experimental setup for this thesis has been immense. His way of explaining concepts in a simple manner and the innumerable discussions that I had with him will always be cherished.

I also extend my thanks to my other friends in ISIP for making my tenure a memorable one. Last but not the least, I would like to thank all my friends who have added so much fun to my life and helped me keep mind off work when needed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

An automatic speech recognition system is a machine that converts the input speech signal to text. The ability of the machine to accept human speech and act accordingly is useful in several scenarios in which speech is the only form of communication. To make such a machine useful and reliable, it should be independent of speaker, language and other disfluency problems associated with speech. The problem of mathematically modeling speech is difficult because some of the mechanisms underlying speech communication are still not clearly understood [1,2]. Hence a statistical approach is used to formulate the speech recognition problem because of the simplicity with which the measured information is self-organized.

The problem of recognizing speech can be formulated mathematically using Bayes theory [1]. The problem is posed as one in which the goal is to find the string of words that were spoken given the acoustic evidence in the form of input features. If W represents the string of words that belong to a predefined vocabulary [1] and the acoustic evidence A is observed, then the solution is to find $\hat{W}$ such that

$$\hat{W} = \arg\max_{W} p(W/A) \tag{1}$$

The recognizer should choose the most likely word sequence given the acoustic evidence and a vocabulary. Using Bayes formula [1, 3], $P(W/A)$ can be written as

$$P(W/A) = \frac{P(A/W)P(W)}{P(A)}, \quad (2)$$

where $P(W)$ is the probability of the string of words W and is usually given by the language model [1]. $P(A/W)$ is the probability of observing the acoustics A given the string of words W. Since the acoustic evidence is fixed with respect to the maximization process, (2) reduces to

$$\hat{W} = \arg\max_{W} p(A/W)p(W). \quad (3)$$

$P(A/W)$ is estimated using an acoustic model which is trained from audio speech data and transcriptions. The focus of this thesis is the estimation process for $P(A/W)$.

1.1. Overview of a Speech Recognition System

A speech recognition system consists of three major components. They are the acoustic front end that converts an input signal to a set of features, an Expectation Maximization (EM) [1,4,5,6] based model trainer and a decoder that generates the final word hypotheses. This section describes these important components of a speech recognition system.

The speech signal and the spectrogram of an utterance *"one one one"* are shown in Figure 1. Since human speech is heavily dependent on context and style of articulation [7,8,9], the waveform and spectrogram for these three examples of the same word are totally different, even for the same speaker. The goal of the acoustic front end is to extract salient information from the input speech signal for better classification. In the

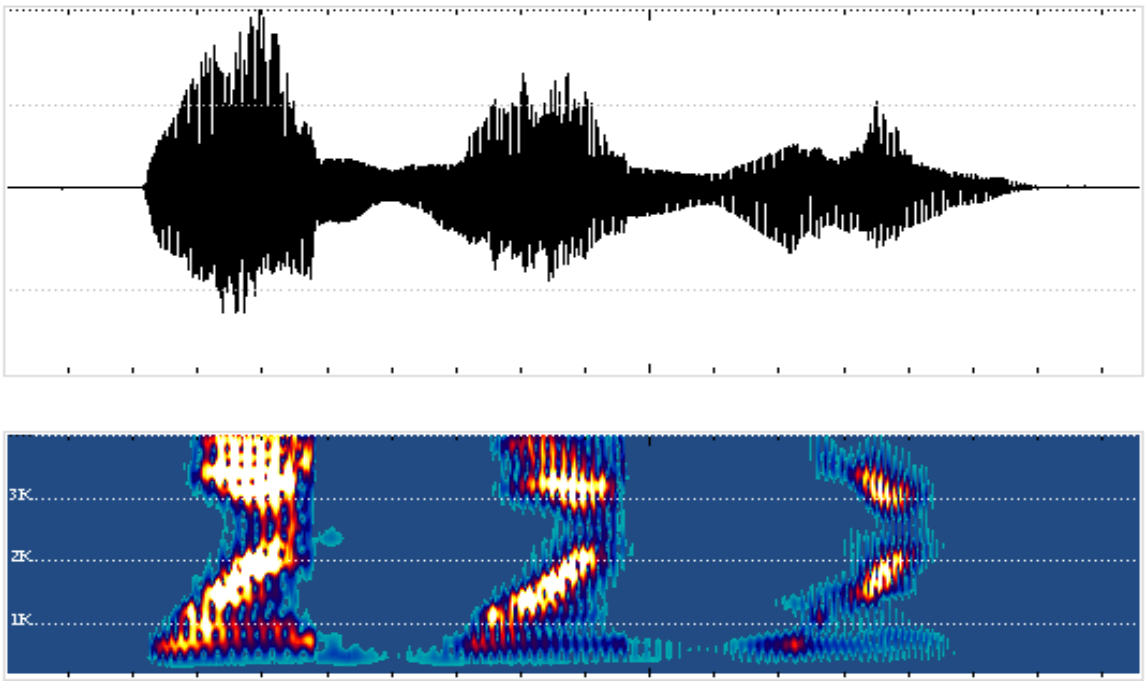

Figure 1. Speech signal and spectrogram for an utterance “one one one.” Note the variation in both the signal and spectrogram for three examples of the same word.

front end, knowledge of human speech perception and speech signal processing techniques [10,11,12] are combined. The front end takes advantage of the stationary characteristics of a speech signal. The signal is typically analyzed using a 10 msec frame duration and windowing is employed to smooth the frame boundary effects [10]. Cepstral coefficients are derived after performing an FFT analysis and a standard mel-scale filter bank [13,14]. The first and second derivatives for these base features are then calculated. The first and second derivatives help capture the temporal evolution of the spectrum that is important for some speech sounds [10,12,13] and help to a certain extent in solving the coarticulation problem [12].

Typically, speech recognition systems generate a 39-dimensional feature vector. Some normalization techniques are also performed to make the features robust to channel

noise and other distortions [15]. Discriminative and representative techniques are applied to the front end for dimensionality reduction and better feature separability [3,16,17]. The spectral information extracted from the front end does not solve all acoustic modeling problems that occur in normal speech (e.g., poor pronunciation, coarticulation) but represents a reasonable compromise between computational resources, complexity, and performance. There are numerous approaches to acoustic modeling that address these remaining problems. Most of these approaches use data-driven techniques [18].

Acoustic models used in speech recognition are typically based on hidden Markov models (HMM) [1,3,4]. The elements of an HMM are:

- the number of states in the model, denoted by N;
- the number of distinct observations for each state, denoted by M;
- a state-transition probability distribution $A=\{a_{ij}\}$;
- an output observation probability distribution $B=\{b_i(k)\}$;
- an initial state distribution π.

The output probability distribution at each state of an HMM is a continuous probability distribution typically modeled as a multivariate Gaussian distribution [1,3,4]. A Gaussian distribution can be written as:

$$b_j(\boldsymbol{o}_t) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_j|}}\exp\left(-\frac{1}{2}(\boldsymbol{o}_t-\mu_j)'\Sigma_j^{-1}(\boldsymbol{o}_t-\mu_j)\right) \quad (4)$$

where n is the dimension of the observation vector $\boldsymbol{o}_t$ and the subscript j denotes the j^{th} state. Models are built by estimating the parameters of a Gaussian distribution in an

iterative fashion using the training data. Phonetic acoustic models [1,4,19] are typically used to keep the number of acoustic models within reasonable limits and provide enough training data for each model. The details on training phonetic models are discussed later in Section 1.3. Acoustic models can also be based on other techniques such as artificial neural networks [20] and dynamic time warping [21].

The decoder finds the best possible word sequence given a set of acoustic models and a set of input features, which we often refer to as observations. The problem of finding the best possible word sequence given a set of observations is generally considered to be a search problem [22,24]. We typically use a breadth-first search technique called Viterbi Search [22,24] where all hypotheses are pursued in parallel. To keep the memory and computations within reasonable limits, different paths are merged using Viterbi pruning [22]. Also, beam pruning techniques [23,24] are used to discard all the hypotheses that fall below a certain threshold of the best scoring hypothesis. To add more knowledge to the search process and to limit the search space, the decoder also uses an N-gram language model [1,25,26,27]. Decoders use several implementation techniques such as lexical trees [24,28], language model lookaheads [24,29], and cached language models [27] to keep the memory requirements and computational resources within reasonable limits.

1.2. Supervised Learning in Speech Recognition

Learning [3] is an algorithmic process of reducing the modeling error on a set of training data. Learning can take two forms: supervised and unsupervised. In supervised

learning, a set of training samples with the corresponding labels are provided. The algorithms for supervised learning seek to increase the probability of the data given the existing model parameters. The models are updated after every pass to obtain better estimates of the model parameters. In unsupervised learning, there are no labels for the training samples and the system forms natural groups of input data by the process of *clustering* [30,31]. The clusters or the grouping may change depending on the clustering criteria and the input data.

In a speech recognition system, the maximization of the probability $P(A/W)$ in (3) defines the process of supervised learning. The problem is to estimate the parameters of the model given a model structure and labeled training data. The parameters of each state of an HMM can be updated if we knew exactly which state sequence produced the input data. But the input state sequence is not known explicitly in an HMM. Hence, an efficient procedure called Baum-Welch training is used to estimate the model parameters [4,32]. This procedure is not computationally intensive. In Baum-Welch training, a forward probability, given by

$$\alpha_t(i) = Pr((O_1, O_2, \ldots, O_t, i_t = q_i)/\lambda), \tag{5}$$

is defined as the probability of the partial observation sequence (until time t) and being in the state q_i at time t, given the model λ.

Similarly, a backward probability is given by

$$\beta_t(i) = Pr(O_{t+1}, O_{t+2}, \ldots, O_T/i_t = q_i, \lambda), \tag{6}$$

which is the probability of the partial observation sequence from $t + 1$ to the end of the utterance, given state q_i at time t and the model λ. Both the forward and backward probabilities can be solved inductively assuming a lattice structure that avoids redundant computations [4]. This efficient implementation is known as the forward-backward algorithm [4,12] and is an integral part of the Baum-Welch training procedure.

The parameters of the Gaussian distribution, namely the mean and the covariance, are reestimated as follows [4,33,34,35]:

$$\hat{\mu}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t) o_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t)}, \tag{7}$$

where $L_{jm}^{r}(t)$ is the state occupancy probability, R is the total number of observations, T is the total duration of each utterance and o_t^r is the observation vector for the t^{th} frame in the r^{th} utterance during the training process. In other words, the probability of being in a particular state j, is calculated across the feature vectors at all possible time instants and each feature vector is weighted by this probability in updating the Gaussian parameters.

The state occupancy probability is given by

$$L_{jm}^{r}(t) = \frac{\alpha_j(t)\beta_j(t)}{P_r}, \tag{8}$$

where P_r is the probability of the utterance and is used as a normalization factor. Similarly, the covariance and the mixture weights are updated as follows

$$\hat{\Sigma}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t)(o_t^r - \hat{\mu}_{jm})(o_t^r - \hat{\mu}_{jm})'}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t)} \tag{9}$$

$$c_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{j}^{r}(t)} \tag{10}$$

According to the EM algorithm, the Baum-Welch reestimation procedure guarantees a monotonic likelihood improvement on each iteration and eventually the likelihood converges to a local maximum. Another training procedure called Viterbi training [36] is also used frequently. Discriminative training methods such as Maximum Mutual Information Estimation (MMIE) [37] and Support Vector Machines [38,39] are gaining popularity and are used in conjunction with existing methods.

1.3. Practical Issues in Training

The theory behind supervised training was discussed in the previous section. However, in order to obtain a good acoustic model, there are several additional stages in the actual training process. These stages include seeding the initial models, training silence, context-independent and context-dependent phone models, and enhancing this models using mixture distributions. The details of each stage are explained in this section. The underlying theory of using the forward-backward procedure to estimate the model parameters remains the same and is used iteratively. Hidden Markov Models are used with

Gaussian mixtures as the underlying distribution. A typical training process, which is often referred to as a recipe, is shown in Figure 2.

To begin the supervised training process, transcriptions should be available for all speech training data. The phone set and the lexicon that maps the words to their corresponding phone-level pronunciations should also be defined. The topology of the acoustic model plays an important role in the overall performance [40] and needs to be engineered. Before the training process is started, parameters of the HMM, namely the mean and variance, need to be initialized. There are several methods for seeding the parameters of HMM [3,35]. One such technique known as flat start [35] involves computing the global mean and variance across all training data, and then initializing all models with this global mean and variance.

In a large vocabulary system, the words are broken into sub-word units called phones and acoustic models are built for each phone. The number of phones used to

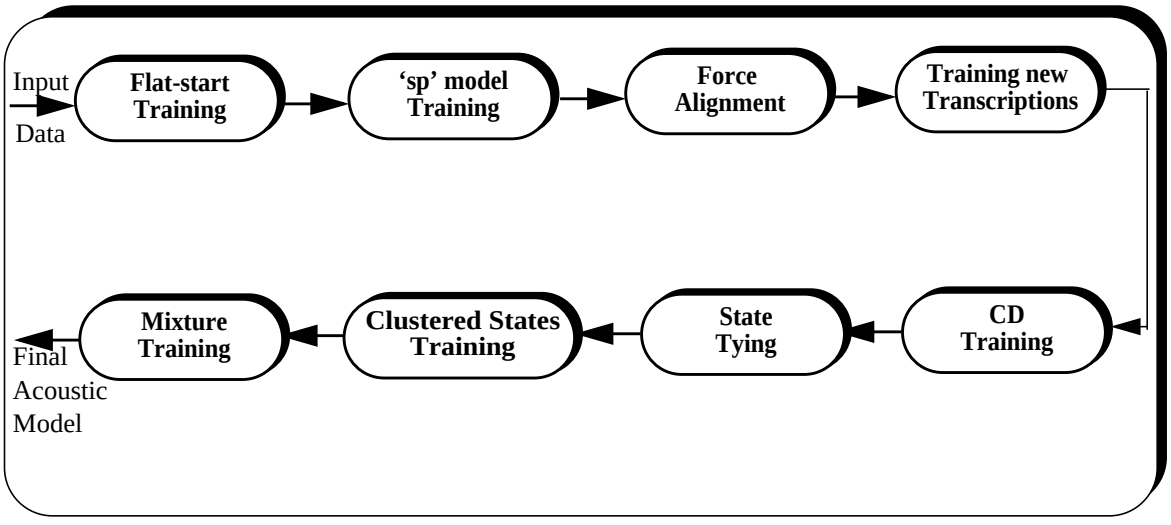

Figure 2. Various stages of the training process starting from flat start to mixture training. The lexicon and the phone set are predefined.

represent the words in a database depends on several factors such as the complexity of the system, amount of training data, etc. Typically for American English, 35 to 45 phones are used. The phone-level transcriptions for monophone training are obtained by subdividing each word into its corresponding phone equivalents. The phone set is predefined and only these predefined phones are used to obtain the phone-level transcriptions. The context information is not used since only monophone training is done.

Examples of context-independent and context-dependent models are shown in Figure 3. In a typical training recipe, context-independent phone models, often referred to as monophone models, are created using a flat-start procedure. Context-dependent models are then bootstrapped from these context-independent models, and trained for several iterations using these phone-level transcriptions.

Many recognition systems use some kind of acoustic model to capture the interword silence [35,41]. In the ISIP-ASR system [24], a short pause model, denoted ‘sp’ is used. This is a 1-state HMM which can be skipped completely if needed. After four

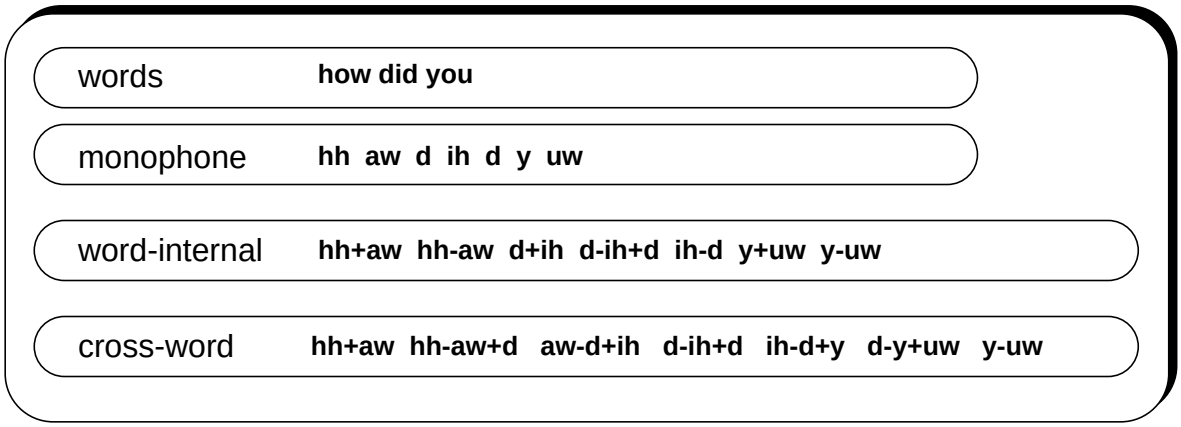

Figure 3. Example of monophone and context-dependent phone realizations for a transcription — “+” denotes right context and “-” denotes left-context.

passes of flat-start training, short-pause training is done. During this stage of training, the short pause model is introduced between each word in the transcription and the training is continued as before. If there is a short pause between words, then the ‘sp’ model will model these interword short silences. Prior to this stage in the training process, silence and short pauses were inserted manually into the input transcriptions (and are inherently inaccurate).

A related problem is that some words can have multiple pronunciations. It is expensive and time-consuming to have linguists manually make decisions about which pronunciation was actually used. Instead, we let the system choose where silence occurs and what pronunciations need to be chosen for a given utterance. This is done by performing running a Viterbi alignment [22] on the training data using word-level transcriptions and a lexicon. This also helps in identifying training data with erroneous transcriptions because these data cannot be aligned properly and are rejected. Once the alignment is done, monophone training is continued using the new set of phone transcriptions given by this alignment process.

In order to train context-dependent models, often referred to as triphone models, context-dependent transcriptions need to be generated [23]. As illustrated in Figure 3, context-dependent transcriptions are generated from the monophone transcriptions. If the contexts across words are taken into account, then cross-word transcriptions are generated. If only the within-word contexts are considered then word-internal transcriptions are generated. After the transcriptions are obtained, context-dependent triphone models are trained. The number of acoustic models that needs to be trained now increases

significantly compared to the monophone stage. There might not be enough training data for all triphone models. Hence, states of different models are tied together so that they can share the same training data. This helps insure that each model has a sufficient amount of training data. This process of sharing training data across states is called state tying [42]. During state tying the states of context-dependent models are tied together based on phonetic contexts using decision trees [42]. The entire process is automated and data-driven, which allows it to be tightly integrated into the recognition process. State tying also helps in generating models that are not present in the training set but can occur in the test set. Once the models have been tied and transformed to context-dependent models, the training process continues as before using standard reestimation techniques.

After the context-dependent models are sufficiently trained, models with multiple Gaussian mixtures per state are generated and trained — a process known as mixture training [43]. Generally, all states have the same number of mixtures per state. The idea behind mixture training is that each mixture component will model a different modality [43,44] in the training data — male and female speakers, different kinds of background noise, etc. The Gaussian mixtures are split [35] by perturbing them around their mean value leaving their variance unchanged. Training is continued by splitting the Gaussians and training them until the required number of mixtures are obtained.

It is not necessary that the above-mentioned training procedure be followed in all applications. The procedure can be altered depending on the complexity of the task, required accuracy and desired computational speed. For some complex databases, only word-internal contexts are used to reduce the memory requirements during recognition.

The training procedure is simplified and systems are made to run in real time for simple tasks like digit recognition where high accuracy has been obtained. For complex tasks such as conversational speech, more rigorous training procedure is followed and complex models are built.

1.4. Thesis Objective and Organization

The primary objective of this thesis is to analyze the performance of a speech recognition system in the presence of mislabeled transcriptions. Several experiments have shown that it is possible to achieve reasonable performance using data with erroneous transcriptions [45,46,47]. But no significant work has been done to analyze why the training algorithms are robust to mislabeled transcriptions. This thesis will explore the reasons behind the robustness of the training algorithms at a fundamental level. The hypothesis of this thesis is that the EM-based supervised training is robust to mislabeled data because the Gaussian distributions that are used to model the data can reject the noisy data present in small quantities.

The thesis is organized in as follows. Chapter 2 describes the experimental design for the thesis. It describes the various experiments that were performed and how these experiments fit into the framework of this thesis. Preliminary results on various speech databases are presented. Chapter 3 provides an analysis of the training process to mislabeled transcriptions. Each stage in the training process is analyzed using a subset of the Alphadigits [48] database. Chapter 4 summarizes the findings from this thesis and discusses some promising avenues for future work.

CHAPTER 2

EXPERIMENTAL PARADIGM

The primary objective of this thesis is to explore the effect of transcription errors on the overall performance of a speech recognition system. It is necessary that different types of transcription errors be introduced in varying amounts to study their effect on the overall performance of the system. This analysis would help categorize the effects of various types of transcription errors based on their impact on recognition performance. Even for the same level of transcription errors, the performance of the system can vary depending upon the complexity of the database and the training procedure used. Hence, experiments were performed on three different databases of different complexities. Some simulated experiments were also performed to better understand the effects of transcription errors on the training process using Gaussian mixtures.

2.1. Corpora

The effect of the transcription errors could be vastly different across different databases. There could be several reasons for such a difference in performance. For example the effect could depend on the vocabulary of the database, the manner in which the original database was segmented or quality of the speech recordings. Experiments were performed on three popular databases: TIDigits [49], OGI Alphadigits [48] and Switchboard [50].

TIDigits database was collected by Texas Instruments in 1983 to establish a common baseline for performance on connected word recognition (CWR) [49] tasks. The database has a vocabulary of eleven words. This includes numbers from ‘zero’ through ‘nine’ and ‘oh’ - an alternate pronunciation for zero. The recording conditions consisted of speech collected in a studio quality recording environment and included over 300 men, women and children. The database has about 6 hours of training data amounting to 12,549 utterances and about 6 hours of data for testing purposes. Word error rates as low as 0.4% have been obtained using word models for training [51].

The Alphadigits (AD) database was collected by OGI [48,52] and the vocabulary includes all letters of the English alphabet as well as the digits — zero through nine. The database has about 54.6 hours of training data and 3.5 hours of test data and includes over 3,000 speakers for training. Alphadigits is a more difficult task than TIDigits because the vocabulary is larger and the recording is not of studio quality. Typically, cross-word triphone acoustic models are trained and loop-grammar decoding [24] is performed for recognition. The error rates are around 10% for clustered triphone acoustic models [24,52].

The most widely used database for large vocabulary conversational speech is the Switchboard (SWB) database collected by Texas Instruments in the early 1990’s [50]. The database was collected using a digital interface to the public telephone system. The data collection scenario involved two people talking to each other on some mutually agreed upon topic. There are 2,438 conversations involving an even mix of male and female speakers. The vocabulary is around 100,000 words. Several factors such as disfluencies in

speech [9], a wide range of speakers, recording conditions and a very large vocabulary make it a difficult task. During the last few years, much improvement has been made in recognizing conversational speech using the Switchboard database [53]. The word error rate is around 25% for state of the art systems in the recent Rich Transcription Evaluations [41,54,55].

The quality of the reference transcriptions has always been an issue, and was a major motivation for this work. In recent years, significant effort has resulted in a reduction in the transcription error rate from approximately 8% WER to less than 1% WER [56]. Non-speech events like background noises, lip smacks, laughter, channel distortions etc. have also been accurately marked in these transcriptions [56,57,58]. Yet, to our surprise, speech recognition error rates have not dropped appreciably when using these improved transcriptions [59]. Understanding this phenomena was a major motivation for this work.

2.2. Introducing Errors

To analyze the performance of a system trained on erroneous transcriptions, transcription errors were introduced into the clean databases. The performance with imperfect transcriptions was then analyzed and compared with training performed using perfect transcriptions. This approach is flexible because the various types of errors can be introduced in a controlled manner.

Before introducing errors into a database, it is necessary to understand the types of errors that can be made when a database is transcribed. There are three different types of

errors possible when a database is transcribed, namely substitutions, deletions and insertions. All these errors are likely while transcribing a database. Substitution errors are generally made when similar sounding words or phones are substituted for the original word. For example, the word "*yeah*" is usually transcribed as "*the*", when the speaker articulates the word poorly. Deletion and insertion errors are typically made with speakers who repeat words or have poor articulation. For example, if the words spoken were "*I I I know she did that*", then it is possible for the transcriber to delete or insert one "*I*" and transcribe it as "*I know she did that*" or "*I I know she did that*" respectively. Another important issue related to transcription of conversational speech is the issue of partial words [56]. Some non-speech events, such as laughter and silence, are not properly identified and transcribed as words.

When the errors were introduced in the database for this thesis, only the substitution, deletion and insertion type errors were introduced. Automated scripts were developed that introduce different types of errors in a controlled fashion (e.g., varying the word error rate and the context in which the error is introduced). Errors were introduced only in word-level transcriptions since speech is mostly transcribed at the word level. If the training database had 10,000 words, then a substitution type transcription error rate of 10% would mean that 1,000 words in the training database would be replaced with incorrect words.

The process of introducing transcription errors is described below. The total number of words in the training database is computed. The total number of words that need to be in error is determined using the target transcription error rate and the total

number of words in the database. The list of unique words in the database is given by a lexicon. The total number of times each word has to be in error is found from the total number of unique words and total number of words that need to be in error. The errors are introduced in two different ways: *equiprobable* and *random*. In *equiprobable* mode, all possible words get an equal weight in corrupting a given word incase of substitution or insertion error. If a word *'one'* needs to be substituted 10 times and if there are 10 other possible words that can replace it, then each word replaces the word *'one'* once in *equiprobable* mode. In *random* mode, a given word is corrupted in a completely random manner by all other possible words. The utterances that are to be corrupted in the database are chosen to span the whole database and all speakers in the database. A combination of these three types of errors can also be introduced in the database. It is possible to corrupt the database at 10% error in which substitution errors are 5%, insertion errors are 3% and deletions contribute 2%.

The process used to introduce errors as discussed above was used for relatively small vocabulary tasks like TIDigits and AD. However for SWB, due to its large vocabulary, the procedure was altered. The complete procedure was randomized. The number of words that needs to be corrupted in the database was calculated as before based on the total number of words in the database and the target transcription error. Also, as before a list of unique words for the database is given by a lexicon. After calculating the number of words that needs to be in error, a word that needs to be in error is chosen at random from the database and the replacement word is also chosen at random from the list of unique words. This is repeated until the target transcription error rate is achieved.

2.3. Experimental Results

As mentioned earlier, experiments were performed on three databases: TIDigits, Alphadigits and Switchboard. For each database, automated scripts were used to corrupt the database by introducing the required type of error at various levels. The errors were introduced in *equiprobable* mode for TIDigits and Alphadigits and in *random* mode for Switchboard. This section describes the various experiments performed for each database.

Experiments for TIDigits were performed on a standard training set of 12,549 utterances and a standard test set of 12,547 utterances [49]. Training was performed using word models to obtain 16-mixture per state Gaussian models. Loop-grammar decoding [24] was done to obtain the final hypotheses. The error rate in the transcriptions was increased in powers of 2 to get transcription errors ranging from 1% to 64%. Baseline system results were obtained using a completely clean set of transcriptions. Experiments were performed by introducing substitution, insertion and deletion type errors. Weighted errors were also introduced in the database to analyze the performance of the system in the presence of combinations of errors. The ratio of different types of errors in the weighted error scheme was 4:3:1 for the insertion, substitution and deletion categories respectively. This ratio was chosen because the error distribution in the baseline system without transcription errors was 4:3:1 for insertion, substitution and deletion errors respectively.

The results are shown in the form of a graph in Figures 4 and 5. The independent variable is the base-2 log of the transcription error rate (TER) while the dependent variable is the word error rate (WER). It can be observed that for a small vocabulary system transcription errors do not make a significant impact even at a 16% transcription

error rate. For the transcription errors to make an effect on the overall performance, they have to be present in high percentages (typically more than 30%). This is true for all types of transcription errors, namely substitutions, deletion, insertion and weighted errors. The same trend can be observed for a 1-mixture system and a 16-mixture system. Both these system perform poorly only at significant but unlikely transcription error rates.

Alphadigits experiments were performed using a standard training set of 51,544 utterances and a test set of 3,329 utterances [52]. For all experiments, 12-mixture state-tied cross-word acoustic models were used. Decoding was performed with a loop grammar. Baseline experiments were performed with a clean set of transcriptions using 1-mixture and the final 12-mixture acoustic models. Only substitution type errors were introduced in the database. Experiments were done with transcription error rates of 2% and 16% respectively and the results were compared with the corresponding baseline systems. The results are shown in Table 1.

Training for SWB was performed using the SWB-I training set [60,61]. This amounted to 60 hours of training data covering 1,925 conversation sides. The test set had 38 speakers and a total duration of 30 minutes. Twelve-mixture state-tied cross-word acoustic models were trained. Decoding was performed using a lattice rescoring mode [24] to generate the final hypotheses. A baseline experiment was performed with a clean set of input transcriptions. Two more experiments were performed by introducing substitution type errors in the database in a completely random manner. The transcription error rates for these experiments were 2% and 16% respectively. The results are also tabulated in Table 1.

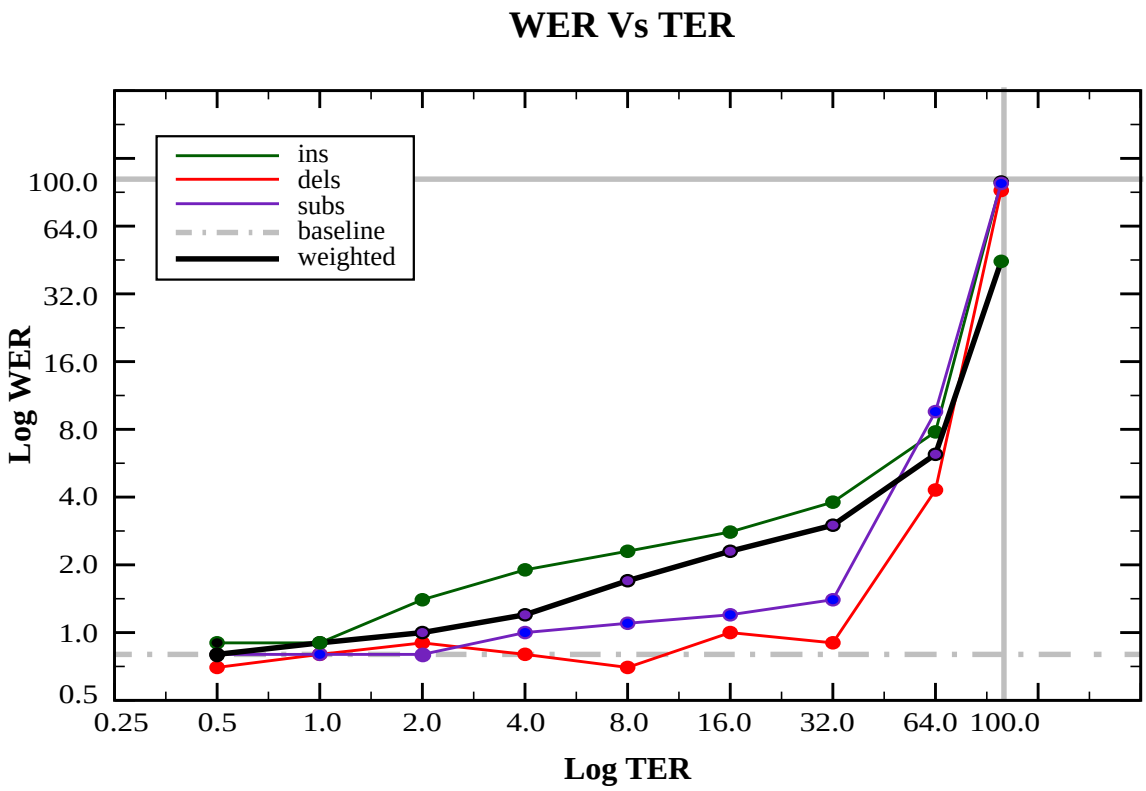

Figure 4. A log-log plot of transcription error rate (TER) vs. word error rate (WER) for experiments performed on the TIDigits database. The pattern is the same for all types of transcription errors in that WER does not increase significantly until the TER is greater than 16%.

It can be observed from Table 1 that the transcription errors do not make a significant impact on any of the databases. Also, as the acoustic model is enhanced using multiple mixture Gaussians per state, the transcription errors have a smaller impact on the recognition performance. Even for a complex database like SWB, the word error rate degrades only by 3.5% (absolute) at a 16% transcription error. These experiments seem to indicate that the training process is robust to transcription errors that are normally present in a database.

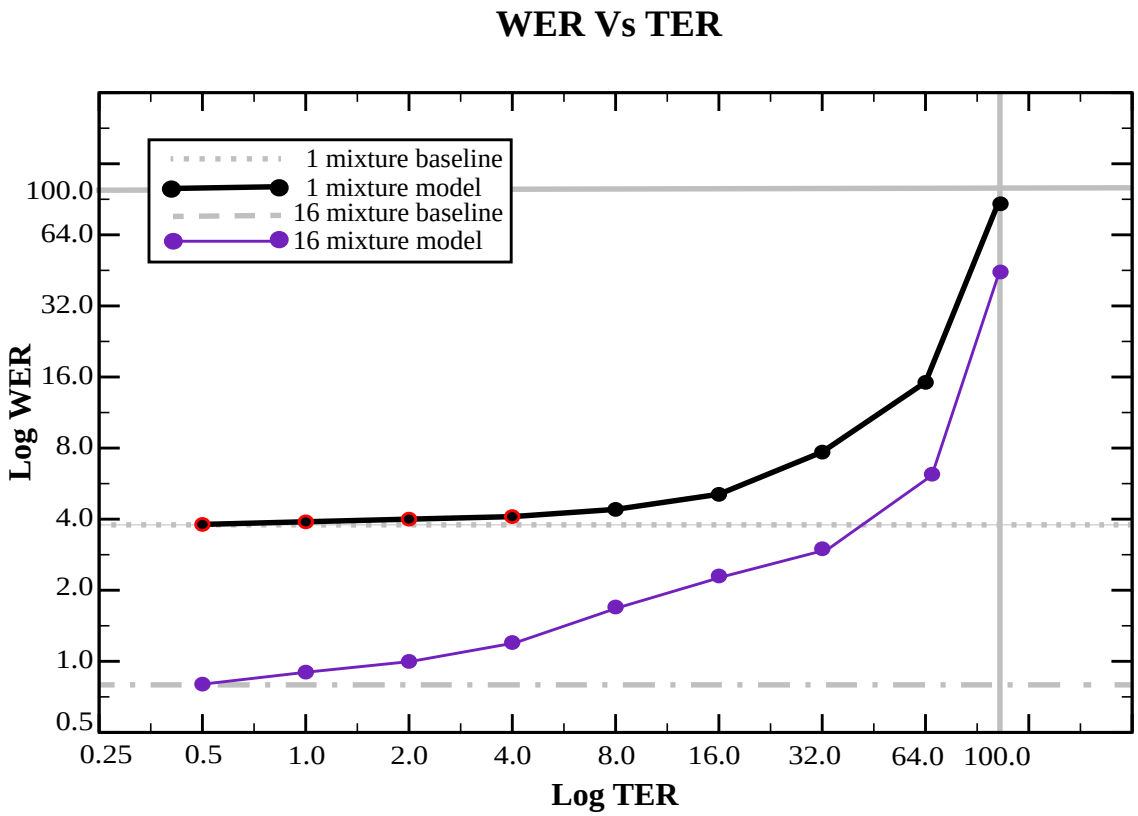

Figure 5. A log-log plot of transcription error rate (TER) vs. word error rate (WER) for experiments performed on the TIDigits database. In this case, 1-mixture and 16-mixture acoustic models are compared. WER again degrades only for TERs above 16%.

2.4. Simulated Experiments

Simulation is a process of designing a model of the real system and performing experiments with this model. Simulated experiments are generally done when the actual experiments cannot be performed due to several constraints [62]. In the case of simulation, it is also possible to control one particular variable and analyze the behavior of the system, which might not be possible in a real system. In the previous section we saw that transcription errors do not degrade the performance significantly. But since the whole

Corpora	Acoustic Models	Transcription Error Rate WER		
		0%	2%	16%
TIDIGITS	1 mixture word	3.8	4.0	5.1
	16 mixture word	0.8	1.0	2.3
Alphadigits	1 mixture xwrd	31.9	32.3	36.2
	16 mixture xwrd	10.8	10.8	12.1
SWB	12 mixture xwrd	41.1	41.8	44.6

Table 1. Comparison of the baseline system (clean transcriptions) with systems trained on transcriptions with substitution errors. At a 16% transcription error rate, the word error rate does not increase significantly compared to the baseline system for the three databases.

process is complex, this robustness to transcription errors cannot be attributed to one single phenomenon. Hence, simulated experiments were performed to better understand this robustness to transcription errors.

An important problem with real speech recognition data is the dimensionality of the space in which recognition is performed [3]. The input feature vectors in a speech recognition system have a dimensionality of more than thirty which is not easy to visualize and the computations are not easily tractable. Hence for easy visualization and tractable computations, simulated experiments were carried out with one dimensional data. It should be easy to extend the results from the one-dimensional data to multidimensional data because the real system is built under the assumption that feature vectors are not correlated [24]. Also in a real system, there are many competing models that add to the overall complexity. A good starting point would be to understand the case

in which there are only two models under consideration and one of the models is corrupted by the data from the other. Using simulated experiments, several variables in the training process, such as the forward and backward probabilities can be eliminated.

The experimental setup for the simulated experiments is discussed below. Two Gaussian distributions were considered, one of them being the original *correct* distribution and the second one being a *corrupting* distribution. These distributions can have arbitrary means and variance. A new distribution is estimated from the data generated from these two distributions. At zero percent transcription error, the data for estimating the parameters of this new distribution is obtained from the original correct distribution. As the transcription error rate is increased, the data for estimating the parameters of the new distribution is obtained from both the correct distribution and the corrupting distribution at required percentages. This is analogous to what happens with imperfect transcriptions when the models learn from incorrect data. The experiment is simplified here by considering only two single-mixture models.

The goal of the simulated experiment is to quantify the effect of erroneous data on the estimated distribution. This can be measured in several ways. A rather simple and straightforward way is to analyze how the mean and variance begin to vary as the error in the data is increased. This would provide a rough estimate of how close the estimated distribution is with respect to the original distribution. Another measure that has been widely used is the Kullback-Leibler [3] distance that measures the divergence between two distributions. For continuous distributions, the K-L distance is given by

$$D_{KL}(p(x),q(x)) = \int_{-\infty}^{\infty} q(x)\ln\frac{q(x)}{p(x)}, \quad (11)$$

where $p(x)$ and $q(x)$ are two continuous distributions. The K-L distance measure is asymmetric, because if $p(x)$ and $q(x)$ are interchanged in (11) then a different distance measure will be obtained. The K-L divergence value will be close to zero if the distributions are similar to each other. Though K-L distance is a good metric, it does not solve the problem from a pattern recognition point of view. There could be a case where for a particular data error rate, the estimate of the original distribution is incorrect but the recognition accuracy may be unaffected. In such a case, the K-L distance measure would suggest that the estimated distribution is not good enough but it is acceptable for classification since recognition is not affected. Hence a more reasonable error metric would be the probability of error [3].

The probability of error measures the theoretical error between two distributions and is given by

$$P(e) = P(x \in \Re_2 | \omega_1) + P(x \in \Re_1 | \omega_2), \quad (12)$$

where $\Re_1$ and $\Re_2$ are the two regions after classification, belonging to the two classes ω_1 and ω_2 and x is the input observation. Equation (12) implies that there are two ways a classification error can occur. The input observation x could be in the region $\Re_2$ while it actually belongs to the class ω_1 or x lies in $\Re_1$ while it actually belongs to ω_2. Thus, the decision region has a great effect on the probability of error. A good decision region would

reduce the probability of error to the minimum value possible, given the parameters of the distribution. Typically, for a binary classification problem using equiprobable single dimensional Gaussian distributions, the decision region is chosen to be the point of intersection of the two distributions as shown in Figure 6. The black and red colored Gaussians are the two distributions corresponding to class ω_1 and ω_2 respectively. The probability of error is calculated using (12) after finding the decision region on the x-axis. Any data point to the left of the decision region is classified as belonging to Class 1. Similarly, any point to the right of the decision region is classified as belonging to Class 2. The probability of error is the minimum for the decision region shown in Figure 6. Any other point on the x-axis would give a larger probability of error value [3].

For the simulated experiments, the new estimated distribution is used to define the decision boundary. This decision boundary is the point of intersection of the estimated distribution and the corrupting distribution. The decision boundary in conjunction with the two original distributions is used to compute the probability of error. This process is shown in Figure 7. The original distribution is represented by a black colored Gaussian and the corrupting distribution is represented by a red colored Gaussian. The decision boundary is found for various percentages of corrupted data and the probability of error is calculated. The idea behind such an experiment is that as the data gets corrupted, the estimate of the original distribution would be inaccurate which leads to an incorrect decision region. Hence, the probability of error would increase. This increase in probability of error is similar to the likely increase in word error rate as the models get corrupted. The estimate of the correct distribution is calculated for various data error rates

and is represented by a blue-colored Gaussian. If the estimate of the original correct distribution is accurate, then the probability of error is minimum. If the estimate of the original correct distribution is inaccurate, then an improper decision region is chosen and the probability of error increases. Figure 7 shows the probability of error for zero percent and twenty percent corrupted data.

Two experiments were performed using the above described simulated setup to determine how acoustically similar and dissimilar phones perform in the presence of transcription error. For acoustically similar phones, the phones ‘b’ and ‘d’ (plosives) were chosen from the AD set. Also, for acoustically dissimilar phones, ‘aa’ and ‘s’ were chosen. Only one dimension was considered for this experiment. The means and variances

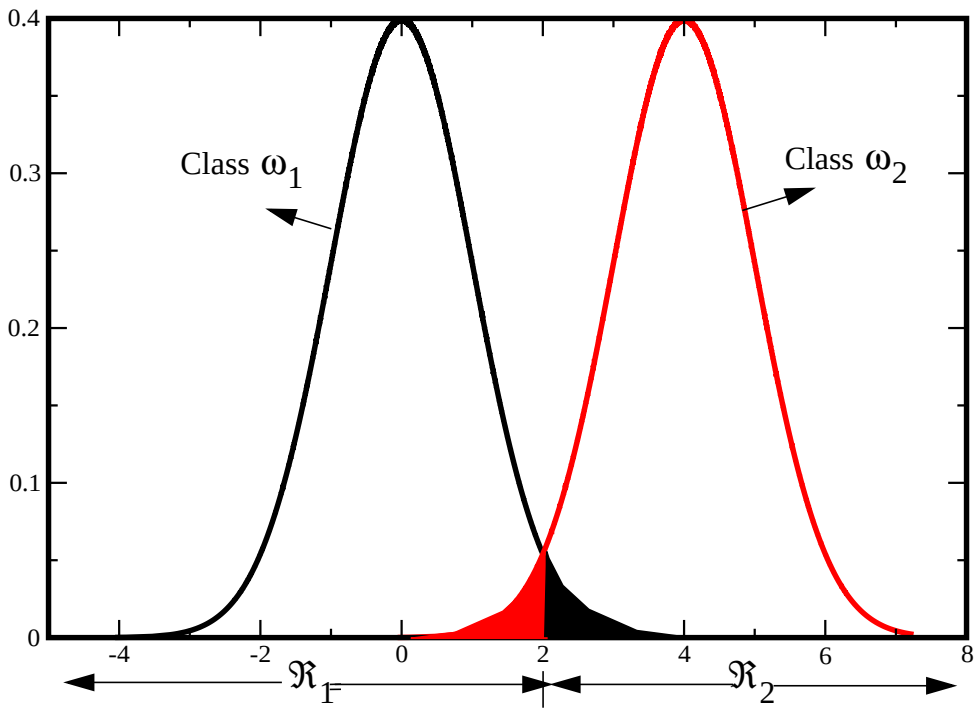

Figure 6. Probability of error between two equiprobable Gaussian distributions. The minimum probability of error is obtained by choosing a threshold that corresponds to the point of intersection between the two distributions.

of the Gaussians were obtained from an AD acoustic model. In the acoustically confusable pair, the original distribution is that of phone ‘b’ and phone ‘d’ is the corrupting distribution with mean values of 0.704 and -0.461 respectively. For the other experiment, phone ‘aa’ is the original distribution and phone ‘s’ is the corrupting distribution with mean values of 4.038 and -5.717 respectively. The transcription error rate was varied from 0% to 20% in steps of two. The results are tabulated in Table 2.

It can be seen in Table 2 that the probability of error is high even at a 0% percent transcription error rate for acoustically similar phones. This is because the distributions for these phones have significant overlap. Also, as the transcription error increases the probability of error does not increase. In the case of acoustically dissimilar phones, the distributions have a small overlap. Hence the probability of error is low at a 0% percent transcription error rate. With the increase in transcription error rate, the probability of error increases but only marginally. This is due to the fact that the Gaussian distributions

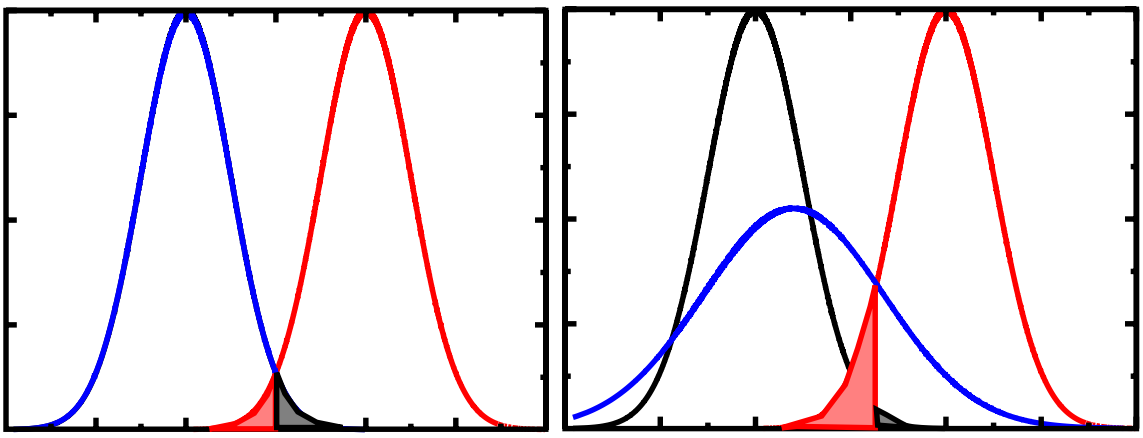

Figure 7. Probability of error calculation for various data error rates. The figure on the left shows the distributions at zero percent data error where the original distribution and the estimated distribution are the same. The figure in the right shows the distributions at 20 percent error where the estimated distribution (in blue) has a wide variance and the probability of error has increased significantly.

Data Error Rate	Probability of Error	
	‘b’ - ‘d’ (acoustically similar pair)	‘aa’ - ‘s’ (acoustically dissimilar pair)
0	44.1	6.84
2	44.1	6.89
4	44.1	7.01
6	44.1	7.12
8	44.1	7.25
10	44.1	7.37
12	44.1	7.49
14	44.1	7.60
16	44.1	7.70
18	44.1	7.79
20	44.1	7.87

Table 2. Probability of error for various transcription error rates on acoustically similar and dissimilar phones. Note that the probability of error does not increase significantly in either case.

tend to cluster around the mean of the data. Hence, even at a 20% transcription error rate, the estimate of the original distribution is not significantly different from the estimate of the original distribution for a 0% transcription error rate. In both the cases we see that the corrupting the model does not increase the probability of error significantly. This is similar to what was observed in the previous section by introducing transcription error in different databases.

In this chapter, the corpora in which the experiments were performed were discussed. The procedure that was used to corrupt each of these databases was discussed in detail. The experiments performed on TIDIGITS, Alphadigits and Switchboard suggest that the transcription errors do not cause a significant degradation in word error rate. To better understand this robustness to transcription errors, simulated experiments were performed in a controlled manner using single dimensional Gaussians and probability of error as an error measure. It was observed that the probability of error does not change significantly with an increase in transcription error rate because the Gaussian models tend to cluster around the mean and need large amounts of erroneous data to cause a significant change in the probability of error.

CHAPTER 3

EXPERIMENTAL ANALYSIS

In chapter 2, it was observed that the transcription errors do not cause any significant degradation in word error rate even at a 16% transcription error rate. The simulated experiments also show that Gaussian probability distributions are adequately robust to model data that is significantly erroneous. In this chapter, we further analyze the effect of transcription errors on the overall acoustic model training process. A small subset of Alphadigits data was chosen for this analysis. Additionally, robustness to erroneous data is analyzed for each stage in the training process.

3.1. Experimental Setup

In chapter 1, we saw that during the training process, the training data is normalized by a value called *state occupancy.* The state occupancy value is used to calculate the model parameters such as the mean and variance during the reestimation process. The mean calculation is given by the following equation

$$\hat{\mu}_{jm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t)\, o_t^r}{\sum_{r=1}^{R}\sum_{t=1}^{T_r} L_{jm}^{r}(t)} \tag{13}$$

where $L_{jm}^{r}(t)$ is the state occupancy probability, R is the total number of observations, T is the total duration of each utterance and o_t^r is the observation vector for the t^{th} frame in the r^{th} utterance during the training process. In other words, the probability of being in a particular state j, is calculated across the feature vectors at all possible times and each feature vector is weighted by this probability in updating the Gaussian parameters.

The state occupancy value can also be defined as the probability of the input data belonging to the model given the current model parameters. The state occupancy values give valuable information about the input data. If the input data matches the model closely, it is likely that the state occupancy value will be high, and the data contributes more to the model reestimation process. On the other hand, if the state occupancy value for the input data is less, then its contribution to the model reestimation process is small. Hence, by comparing the state occupancy values for the correct data (data without transcription errors) and incorrect data (data with transcription errors), it is possible to evaluate the contribution of the incorrect data to the model reestimation process.

To analyze the robustness of the training process in the presence of transcription errors, a subset of Alphadigits database was used. This subset consisted of 4,884 utterances chosen at random. From this set, 100 utterances that had the word ‘o’ were chosen. In these 100 utterances, the word ‘o’ was replaced with the word ‘i’. Then utterances with the original correct transcriptions were added back so that the subset now had 4,984 utterances and a transcription error rate of 7.8%. The motivation for substituting the word ‘o’ with the word ‘i’ is that both words have one phone, ‘ow’ and ‘ay’

respectively, in their pronunciations. Hence a substitution at the word level is equivalent to a substitution at the phone level without any insertions or deletions. The same experimental setup was used to analyze all the stages in the training process. The acoustic models are three-state HMMs with self-loops and transitions to the next state. In each state, Gaussian mixtures were used to model the underlying distribution.

The experiments were performed to verify the following hypotheses:

- How much does the incorrect model learn from the erroneous data during the training process when transcription errors occur?

  This situation is analyzed by observing the state occupancy values of the incorrect model (model ‘ay’ that occurs in the utterance with transcription error) and comparing it with the state occupancy values of the correct model (equivalent model ‘ow’ that occurs in the same utterance but with no transcription errors). If the state occupancy values are low for the incorrect model, then it implies that the incorrect model learns less from the erroneous data when compared to the correct model.

- How much does the erroneous portion of the data contribute to the model reestimation process?

  This is analyzed by observing the state occupancies of the incorrect model (model ‘ay’) in the utterances with transcription errors and comparing it with the state occupancies of the same model in other utterances without transcription errors. If the state occupancy values are low for the erroneous portion of the data then it implies that their contribution to the model reestimation process is low.

3.2. Flat Start And Monophone Training

The initial experiments that were performed on various databases (refer to Section 2.3) did not show any significant degradation in performance in the presence of transcription errors. Hence, the hypothesis is that the state occupancy values for the frames with erroneous data are very low and do not contribute to the model reestimation process.

To verify this hypothesis, the state occupancy for the center state of the phone ‘ay’ was observed for the incorrect utterances (the utterances in which the word ‘o’ was replaced with the word ‘i’). Similarly, the state occupancy for the center state of the phone ‘ow’ was also observed for the correct utterances (the 100 correct utterances that were added later to the list). The state occupancies were analyzed for all iterations of flat start and monophone training. Also, the state occupancy values were normalized by the number of frames for which their values were greater than zero. The normalized state occupancy values for the center state of the model ‘ay’ and ‘ow’ corresponding to the incorrect and correct utterances is shown for all stages of flat start and monophone training in Table 3.

It can be seen that the state occupancy values for the correct center state (corresponding to the model ‘ow’) are significantly higher than that of the incorrect center state (corresponding to the model ‘ay’). Also, it was observed that the number of frames for which the state occupancies were greater than zero is significantly more for the correct state than for the incorrect state. In the utterances with transcription errors, the erroneous data typically gets mapped to the silence model. This shields the center state of the ‘ay’ model from the erroneous data. The incorrect data that occurs when ‘ay’ is substituted for ‘ow’, is mostly rejected during the training process due to its low state occupancy value. Hence, the model learns very little from the incorrect data.

To verify how much the erroneous data contributes to the reestimation of the model (‘ay’ in this case), the state occupancy of the center state of the model ‘ay’ was analyzed from 275 correct utterances (without any transcription error). The state occupancy for the center state of ‘ay’ in these 275 correct utterances was observed to be

0.53 after normalization while the state occupancy of ‘ay’ from the incorrect utterance is 0.148. This shows that the incorrect data does not contribute to the overall reestimation process significantly since its weights are low.

3.3. Context-Dependent Training

Context-dependent training is performed after the monophone models are completely estimated. In this section, we analyze the effect of context dependency and data sharing via state tying on the training process in the presence of transcription errors.

Iteration	Center State of ‘ow’	Center State of ‘ay’
1	0.037	0.037
2	0.122	0.057
3	0.355	0.078
4	0.590	0.150
5	0.633	0.150
6	0.634	0.173
7	0.641	0.159
8	0.639	0.153
9	0.660	0.143
10	0.655	0.153
11	0.659	0.155
12	0.660	0.151

Table 3. Average state occupancy values for the center state in the model ‘ow’ in the correct transcriptions and the model ‘ay’ in the incorrect transcriptions during monophone training. The state occupancy values are higher for the correct transcription. This difference widens after each iteration

As in monophone training, one would expect the state occupancy values to be low for incorrect transcriptions and hence not contribute significantly to the reestimation process. But in the case of context-dependent training, each context-dependent model gets a smaller amount of training data compared to the monophone models. Hence, the percentage of incorrect data the model sees is likely to increase. It is possible that the incorrect data contributes more to the reestimation process and the models can become corrupted.

The cross-word model ‘sil-ay+ey’ was chosen for analysis from the cross-word transcriptions. This model occurs 25 times in the chosen set of utterances of which 4 occurrences were due to the transcription errors introduced earlier as described in Section 3.1. This amounts to a 16% transcription error for this triphone model. Another model ‘f-ay+eh’ was also considered for analysis. This model occurs only three times, and two of these occurrences were due to transcription errors. Hence, this model has a 66% transcription error at the start of context-dependent training.

The state occupancy for the center state for the model ‘sil-ay+ey’ is observed for both correct and incorrect transcriptions. This is done for the four iterations of context-dependent training before state tying. The state occupancy values are tabulated in Table 4. The state occupancy for the correct state stabilizes at 0.57 after 4 iterations. The state occupancy values for the state in the incorrect transcriptions increases after every iteration. The reason behind high state occupancy values for the state in the correct transcriptions is that the context-dependent models were seeded from well-trained monophone models. The state occupancy values for the state occurring in the correct

transcriptions are significantly higher when compared to the state occurring in the incorrect transcriptions even during the first iteration. But due to a relatively high transcription error rate (16% in the case of the model ‘sil-ay+ey’), the state occupancy values increase after every iteration for the state in the incorrect transcription. However, this is insufficient to corrupt the model reestimation process.

During state tying the states of context-dependent models are tied together based on several conditions which are estimated in a data-driven framework [42]. The state-tying mechanism attempts to increase the amount of training data for each context-dependent model. The transcription errors for the models can change depending on the actual data that was shared. If the amount of correct data that is shared outweighs the incorrect data then the transcription errors decrease. This would in turn result in the state occupancy values for the states occurring in the incorrect transcriptions to decrease. Hence, the model

Iteration	Average State Occupancy for Correct Transcriptions	Average State Occupancy for Incorrect Transcriptions
1	0.5223	0.0794
2	0.5808	0.0871
3	0.5827	0.1201
4	0.5772	0.1461

Table 4. Average state occupancy values for the model ‘sil-ay+ey’ during context-dependent training before state tying. The average state occupancy value for the model in the correct transcriptions is significantly more than those in the incorrect transcriptions.

would be less corrupted during the reestimation process as a result of state tying. The following analysis is performed to evaluate the above hypothesis.

After state tying is performed, the center state of ‘sil-ay+ey’ is shared with other models. This increases the number of instances of correct data for this model from 25 to 190 while the number of incorrect instances increases from 4 to 10. The transcription error rate for the model ‘sil-ay+ey’ is reduced to 0.05%. After state tying, 5 more iterations of training were performed and the state occupancies were observed for the center state occurring in the correct and incorrect transcriptions. The results are tabulated for the model ‘sil-ay+ey’ in Table 5.

Table 5 shows that the state occupancy value reduces after each iteration for the center state of the model ‘sil-ay+ey’ in the incorrect transcriptions. It can also be seen that the state occupancy value for the state occurring in the correct transcriptions increases

Iterations	Average State Occupancy for Correct Transcription	Average State Occupancy for Incorrect Transcription
1	0.5829	0.1490
2	0.5807	0.0851
3	0.5913	0.0873
4	0.5915	0.0873
5	0.5910	0.0876

Table 5. Average state occupancy values for the model ‘sil-ay+ey’ during context-dependent training after state tying. The transcription error rate is reduced from 16% to 0.05% by performing state tying.

after each iteration. This is because the transcription error reduces after state tying and the model is now exposed to more clean data than it was before state tying. Hence, the model effectively rejects the incorrect data better than it did before state tying. The state occupancy value for the center state in the incorrect transcriptions decreases drastically after the first iteration and stabilizes after that at 0.08.

The model ‘f-ay+eh’ was also used to verify the hypothesis that state tying improves robustness to incorrect transcriptions. This context-dependent model had a transcription error of 66% before state tying. Before state tying, the average state occupancy value for this model in the incorrect transcriptions was 0.56. State tying significantly decreases the effective transcription error for this model. The state occupancies for the center state of the model ‘f-ay+eh’ in incorrect transcriptions are shown in Table 6. The state occupancy value decreases rapidly from 0.56 before state-tying to 0.16 after 5 passes of reestimation. This shows that state tying adds robustness to the training process by decreasing the transcription error and preventing the models from getting corrupted.

3.4. Mixture Training

To analyze the effect of transcription errors on multiple Gaussian mixtures per state, an experimental setup similar to monophone and triphone training was used. The idea behind multi-mixture Gaussians per state is that each Gaussian mixture component can model the variations in the training data. One Gaussian mixture in a state can model the erroneous portion of the data for that model. If this were to happen then the state

Iterations	Average State Occupancy in Incorrect Transcriptions
1	0.3246
2	0.2020
3	0.2059
4	0.1726
5	0.1621

Table 6. Average state occupancy values for the model ‘f-ay+eh’. The state occupancy values decrease from 0.56 before state tying to 0.16 after five passes of training.

occupancy values would increase even for the incorrect portion of the data since at least one Gaussian mixture component would closely match the data. On the other hand, if there are several modalities in the correct portion of the data, then the incorrect portion of the data would be further rejected and hence have low state occupancy values. This hypothesis is verified in the following analysis.

In order to verify this hypothesis, the state occupancy for the center state of the model ‘sil-ay+ey’ was observed for the correct and incorrect transcriptions. The results are tabulated in Table 7. It can be seen from the table that the state occupancy values for the states in the incorrect transcriptions are again lower than that for the center state in the correct transcriptions. Also, the state occupancy values for the center states in incorrect transcriptions decreases as the number of mixtures is increased. This is because the initial estimates for the Gaussian mixtures are chosen from well-trained single mixture models. Also, during the mixture splitting process, only the mean is perturbed and the variance of the original Gaussian is left unchanged. This results in peaky models even during the

Training Stage	State occupancy in correct transcriptions	State Occupancy in incorrect transcriptions
After 1mixture	0.5372	0.1488
After 2mixture	0.5384	0.1404
After 4 mixture	0.5644	0.1282

Table 7. Average state occupancy values for the model ‘sil-ay+eh’ after each stage of mixture training.

beginning of the mixture training process. This means that the correct portion of the data gains more prominence even during the first pass of mixture training. As the number of mixtures is increased, the model tries to capture all the modalities in the correct portion of the data since it is present in large quantities. Thus the incorrect data is rejected in most of the cases and does not gain any prominence in any of the mixtures.

3.5. Conclusions

From the analysis performed in this chapter, it can be seen that the transcription errors do not corrupt the acoustic models significantly. This is primarily due to the fact that the Gaussian mixtures that are used to model the underlying distribution need a large amount of incorrect data to get corrupted. But since the incorrect data is usually present in very small amounts compared to the correct data, the models are not corrupted significantly. This leads to the effective rejection of incorrect data. The process of iteratively training the models also adds more robustness to the acoustic models. Also, the process of state tying helps in reducing transcription errors by sharing data across different states. This is particularly helpful when the amount of incorrect data tends to increase at

the start of context-dependent training. As the number of mixtures is increased, the incorrect portion of the data is further rejected since each mixture tries to capture the variations in the correct portion of the data and none of the mixtures components model the incorrect data.

CHAPTER 4

CONCLUSIONS AND FUTURE WORK

The previous chapters of this thesis analyzed the effects of transcription errors on the accuracy of a speech recognition system. The training procedure and practical issues in training a speech recognition system were discussed in detail. Experiments performed on different corpora suggest that transcription errors do not cause severe degradation in the performance of a recognition system. This is primarily due to the fact that the Gaussian distributions tend to cluster around the majority of the correct data and the outliers (incorrect data) do not contribute much to the reestimation process. It was also observed that the algorithms used for training give lower weight to the mislabeled data, thereby reducing their contribution to the acoustic model estimates significantly.

4.1. Thesis Contribution

This thesis has explored the robustness of training algorithms to mislabeled data at a fundamental level. This is done by analyzing different types of transcription errors on three different databases: TIDigits, Alphadigits and Switchboard. For Alphadigits, at a 2% transcription error rate, the performance of the system was not affected. With 16% of the data mislabeled, the performance of the system degrades by 12% relative to the baseline results. For a complex task like Switchboard, at 16% mislabeled training data, the performance of the system degrades by 8.5% relative to the baseline results.

The work presented in this thesis also explores the robustness of the training algorithms in the presence of mislabeled transcriptions at a probabilistic level. This was done by analyzing the state occupancies of the correct and mislabeled data at every stage of the training process. The results indicate that it is not necessary to have a very clean database for training. The startup cost of training a system can be reduced and the amount of training data can be increased by using other source of transcriptions such as closed-caption data [46,47,63].

4.2. Experimental Setup and Results

Experiments for this thesis were performed by introducing errors into the three databases. Automated scripts were developed which introduce errors in the corpora in a controlled fashion. The initial experiments on these databases show that the transcription errors do not degrade the performance of the system. To simplify the computations and for easy visualization, simulated experiments were performed using one-dimensional data. These experiments indicate that the Gaussian distributions that are used to model the data are robust to mislabeled data. In other words, they reject the outliers (mislabeled data) present in small quantities compared to the correct data.

Further experiments were performed, as described in Chapter 3, to understand the effects of transcription errors on the overall acoustic model training process. A small subset of Alphadigits data was used for these experiments. Every stage of the training process was analyzed. The state occupancy values are very low for the mislabeled data when compared to the correct data. Hence, the erroneous data does not have a significant

contribution in the model reestimation process. Also, the process of state tying adds robustness to the overall training process by sharing states across the models. This helps in the increasing the amount of correct data during context-dependent training, thereby further reducing the contribution of the mislabeled data to the model reestimation process.

4.3. Future Work

Though the best performance from a system is obtained by using a clean set of transcriptions, the results of this thesis have proven that highly accurate transcriptions are not essential for training an acoustic model. It is possible to closely match the performance of such a system by using other sources of transcriptions such as closed captions, provided there is ample data to overcome the deficiencies of the transcriptions. It would be interesting to quantify how much of these other sources of data are required to match a clean set of transcriptions in terms of system performance. For example, the system could be 90% accurate using 10 hours of clean training data on a database of interest. It is possible that this performance can be matched by using a significantly larger amount of noisy data. Quantifying the exact amount of noisy training data needed to match the performance of clean training data can be an interesting research area to explore in the future.

The experiments performed in this thesis have shown that the Gaussian distributions are more robust to erroneous data because they tend to cluster around large quantities of clean data. Since the mislabeled training data are present in small quantities, the Gaussian distribution rejects them as outliers and this adds to the robustness to the overall training process. Another interesting topic for future research would be to analyze

whether the training procedure is equally robust when using non-Gaussian statistical models such as Laplacian distributions [64,65] to model the data.

REFERENCES

[1] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, Massachusetts, USA, 1997.

[2] N. Deshmukh, A. Ganapathiraju, J. Picone, "Hierarchical Search for Large Vocabulary Conversational Speech Recognition," *IEEE Signal Processing Magazine*, vol. 1, no. 5, pp. 84-107, September 1999.

[3] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification and Scene Analysis*, Wiley Interscience, 2000.

[4] L. R. Rabiner, B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1993.

[5] A. P. Dempster, N. M. Laird, D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1-38, 1977.

[6] C. F. J. Wu, "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics*, vol. 11, no. 1, pp. 95-103, 1983.

[7] K. F. Lee, "Context-dependent Phonetic Hidden Markov Models for Speaker-independent Continuous Speech Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-38, pp. 599-609, April 1990.

[8] J. W. Butzberger, et. al., "Spontaneous speech effects in large vocabulary speech recognition applications," *Proceedings of DARPA Speech and Natural Language Workshop*, Morgan Kaufmann, pp. 339-343, 1992.

[9] E. Shriberg, "Disfluencies in SWITCHBOARD," *Proceedings of the International Conference on Spoken Language Processing*, vol. Addendum, pp. 11-14, Philadelphia, Pennsylvania, October 1996.

[10] J. Picone, "Signal Modeling Techniques in Speech Recognition," *IEEE Proceedings*, vol. 81, no. 9, pp. 1215-1247, September 1993.

[11] V. Mantha, R. Duncan, J. Zhao, J. Picone, "Implementation and Analysis of Speech Recognition Front-ends," *Proceedings of the IEEE Southeastcon*, Lexington, Kentucky, USA, March 1999.

[12] J. R. Deller, J. G. Proakis, J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan Publishing, New York, USA, 1993.

[13] S. Furui, “Cepstral Analysis Technique for Automatic Speaker Verification,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 254-272, 1981.

[14] J. G. Proakis, D. G. Manolakis, *Digital Signal Processing - Principles, Algorithms and Applications,* Prentice Hall, New Jersey, 1996.

[15] R. Heab-Umbach, et. al., “Acoustic Modeling in the Philips Hub-4 Continuous-Speech Recognition System,” *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, Virginia, USA, February 1998.

[16] G. Saon, M. Padmanabhan, R. Gopinath, S. Chen, “Maximum likelihood discriminant feature spaces.”, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. II1129-II1132, Beijing, China, 2000.

[17] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, New York, New York, USA, 1990.

[18] H. Ney, D. Mergel, A. Noll, A. Paesler, “Data Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition,” *IEEE Transactions on Signal Processing*, vol. 40, no. 2, pp. 272-281, February 1992.

[19] K. F. Lee, “Context-dependent Phonetic Hidden Markov Models for Speaker-independent Continuous Speech Recognition,” *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. ASSP-38, pp. 599-609, April 1990.

[20] M. Richard, R Lippmann, “Neural Network Classifiers Estimate Bayesian A Posteriori Probabilities,” *Neural Computation*, vol. 3, no. 4, pp. 461-83, 1991.

[21] L. R. Rabiner, A. E. Rosenberg, S. E. Levinson, “Considerations in Dynamic Time Warping Algorithm for Discrete Word Recognition,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-26, no. 6, pp. 575-582, December 1978.

[22] A. J. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm,” *IEEE Transactions on Information Theory,* vol. IT-13, pp. 260-269, April 1967.

[23] N. Deshmukh, A. Ganapathiraju, J. Picone, “Hierarchical Search for Large Vocabulary Conversational Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 1, no. 5, pp. 84-107, September 1999.

[24] K. F. Lee, *Large Vocabulary Speaker Independent Continuous Speech Recognition*, Ph. D. Thesis, Carnegie Mellon University, Pittsburgh, USA, 1988.

[25] H. Ney, U. Essen, R. Kneser, “On Structuring Probabilistic Dependencies in Stochastic Language Modeling,” *Computer Speech and Language*, vol. 8, no. 1, pp. 1-38, 1994.

[26] M. Jardino, “Multilingual Stochastic N-gram Class Language Models,” *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 161-163, May 1996.

[27] R. Kuhn, R. D. Mori, “A Cache Based Natural Language Model for Speech Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 570-583, 1992.

[28] H. Murveit, P. Monaco, V. Digilakis, J. Butzburger, “Techniques to Achieve an accurate real-time, large vocabulary speech recognition system,” *Proceedings of the ARPA Human Language Technology Workshop*, pp. 368-373, Austin, Texas, USA, March 1995.

[29] H. Ney, S. Ortmanns, “Dynamic Programming Search for Continuous Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 1, no. 5, pp. 64-83, September 1999.

[30] P. Arabie, L. J. Hubert, G. D. Soete, *Clustering and Classification*, World Scientific, River Edge, New Jersey, 1998.

[31] A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[32] L. E. Baum, “An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Function of Markov Process,” *Inequalities*, vol. 1, pp. 1-8, 1972.

[33] J. Picone, “Continuous Speech Recognition Using Hidden Markov Models,” *IEEE Acoustics, Speech and Signal Processing Magazine*, vol. 7, no. 3, pp. 26-41, July 1990.

[34] B. H. Juang, L. R. Rabiner, “Issues in Using Hidden Markov Models for Speech Recognition,” *Advances in Signal Processing*, pp. 509-554, New York, New York, 1992.

[35] P. Woodland, et. al., *HTK Version 1.5: User, Reference and Programmer Manuals*, Cambridge University Engineering Department and Entropic Research Laboratories Inc., 1995.

[36] V. V. Digalakis, M. Ostendorf, J. R. Rohlicek, “Fast Algorithms for Phone Classification and Recognition Using Segment-based Models,” *IEEE Transactions on Signal Processing*, vol. 40, pp. 2885-2896, 1992.

[37] P. Woodland, D. Povey, “Very Large Scale MMIE Training for Conversational Telephone Speech Recognition,” *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, Maryland, USA, May 2000.

[38] C. J. C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *http://svm.research.bell-labs.com/SVMdoc.html*, AT&T Bell Labs, November 1999.

[39] A. Ganapathiraju, J. Hamaker, J. Picone, “A Hybrid ASR System Using Support Vector Machines,” *International Conference of Spoken Language Processing*, Beijing, China, October 2000.

[40] J. Hamaker, *Automatic Learning of Hidden Markov Model Topologies Using Information Theoretic Approaches*, Master of Science Special Project Presentation, Mississippi State University, April, 2000.

[41] A. Martin, M. Przybocki, “The 2001 NIST Evaluation for Recognition of Conversational Speech Over the Telephone,”, *Proceedings of the 2001 Speech Transcription Workshop*, Gaithersburg, Maryland, USA, May 2001.

[42] S. J. Young, P. C. Woodland, “State Clustering in HMM-based Continuous Speech Recognition,” *Computer Speech and Language*, vol. 8, no. 4, pp. 369-384, 1993.

[43] V. Digilakis, P. Monaco, H. Murveit, “Genomes: Generalized Mixture Tying in Continuous HMM-based Speech Recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 4, pp. 281-289, 1996.

[44] X. Huang, A. Acero, H. Hon, *Spoken Language Processing*, Prentice Hall PTR, 2001.

[45] T. Kemp, A. Waibel, "Unsupervised Training of a Speech Recognizer Recent Experiments," *Proceedings of ESCA Eurospeech'99*, pp. 2725-2728, Budapest, Hungary, September 1999.

[46] L. Lamel, J. L. Gauvain, G. Adda, "Lightly Supervised Acoustic Model Training," *Proceedings of the ISCA ITRW ASR2000*, Paris, France, September 2001.

[47] G. Zavaliagkos, T. Colthurst, "Utilizing Untranscribed Training Data to Improve Performance," *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, Landsdowne, Virginia, February 1998.

[48] R. Cole, et. al., "Alphadigit Corpus," *http://www.cse.ogi.edu/CSLU/corpora/alphadigit*, Center for Spoken Language Understanding, Oregon Graduate Institute, Oregon, USA 1997.

[49] R. Leonard, "A Database for Speaker-Independent Digit Recognition," *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, vol. 9, pp. 328-331, March 1984.

[50] J. J. Godfrey, et. al., "SWITCHBOARD: Telephone Speech Corpus for Research and Development," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 517-520, San Francisco, California, March 1992.

[51] F. Zheng, J. Picone, "Robust Low Perplexity Voice Interfaces," *http://www.isip.msstate.edu/publications/reports/mitre_robust/2001/*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, USA, May 2001.

[52] J. Hamaker, et. al., "A Proposal for a Standard Partitioning of the OGI AlphaDigit Corpus," available at *http://isip.msstate.edu/projects/lvcsr/recognition_task/alphadigits/data ogi_alphadigits/trans_eval.text*.

[53] B. Peskin, et. al., "Progress in Recognizing Conversational Telephone Speech," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1811-1814, Munich, Germany, April, 1997.

[54] D. Pallett, et. al., "Overview: Speech Transcription Workshop," *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, Maryland, USA, May 2000.

[55] P. C. Woodland, et. al., "CU-HTK April 2002 Switchboard System," *http://svr-www.eng.cam.ac.uk/reports/svr-ftp/woodland_rt02.pdf*, Cambridge University Engineering Department, May 2002.

[56] "Switchboard Resegmentation," *http://www.isip.msstate.edu/projects/switchboard/index.html*, Institute for Signal and Information Processing, Mississippi State University, Mississippi State, Mississippi, USA.

[57] J. Hamaker, N. Deshmukh, A. Ganapathiraju, J. Picone, "Resegmentation and Transcription of the SWITCHBOARD Corpus," *Speech Transcription Workshop*, Linthicum Heights, Maryland, USA, September 1998.

[58] "Switchboard Transcription Project," *http://www.icsi.berkeley.edu/real/stp/*, International Computer Science Institute, University of California, Berkeley, California, USA.

[59] A. Lolije, et. al., "The AT&T LVCSR-2000 System," *Proceedings of the 2000 Speech Transcription Workshop*, University of Maryland, Maryland, USA, May 2000.

[60] A. Ganapathiraju, et. al., "Syllable - A Promising Recognition Unit for LVCSR," *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 207-214, Santa Barbara, December 1997.

[61] A. Ganapathiraju, et. al., "WS97 Syllable Team Final Report," *Proceedings of the 1997 LVCSR Summer Research Workshop*, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, Maryland, December 1997.

[62] J. P. C. Kelijnen, W. von Groenendaal, *Simulation: A Statistical Perspective*, John Wiley, New York, New York, 1992.

[63] P. Placeway, J. Lafferty, "Cheating with Imperfect Transcripts," *Proceedings of the International Conference on Speech and Language Processing*, vol. 4, pp. 2115-2118, 1996.

[64] R. Heab-Umbach, H. Ney, "Linear Discriminant Analysis for Improved Large Vocabulary Speech Recognition," *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 13-16, San Francisco, California, March 1992.

[65] S. Gazor, W. Zheng, "Speech Probability Distribution," *IEEE Signal Processing Letters*, vol. 10. no. 7, July 2003.