(76)

UTTERANCE VERIFICATION IN LARGE VOCABULARY SPOKEN LANGUAGE UNDERSTANDING SYSTEM

by

HUAN YAO

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 22, 1998

[June 1998]

Author _____

Department of Electrical Engineering and Computer Science
May 22, 1998

Certified by _____

Gregory W. Wornell
Associate Professor of Electrical Engineering
Thesis Supervisor

Certified by _____

Richard. C. Rose
Principal Member of Technical Staff, AT&T Labs – Research
Thesis Supervisor

Accepted by _____

Arthur C. Smith
Chairman, Department Committee on Graduate Theses

UTTERANCE VERIFICATION IN LARGE VOCABULARY SPOKEN LANGUAGE UNDERSTANDING SYSTEM

by

Huan Yao

Submitted to the Department of Electrical Engineering and Computer Science

May 22, 1998

In Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

This thesis develops and evaluates a set of utterance verification (UV) techniques as part of a large vocabulary spoken language understanding (SLU) system. The motivations are to detect out-of-domain utterances, disfluencies, and noises, and also to facilitate confirmation and rejection strategy in dialog control. A two-pass UV procedure is developed. First, speech utterances are decoded by a continuous speech recognizer. Then, a second stage UV mechanism assigns each decoded word a likelihood ratio (LR) based confidence measure computed from subword level LR scores using a set of subword specific hidden Markov models (HMM) dedicated for UV. A discriminative training procedure based on a gradient descent algorithm is developed to estimate the UV model parameters to optimize a cost function that is directly related to the LR criterion used in UV. The verification and the training techniques are evaluated on utterances collected from a highly unconstrained large vocabulary spoken language understanding task performed over the public telephone network. The UV performance is evaluated in terms of the system's ability to accept correctly decoded words while rejecting incorrectly decoded ones.

Thesis Supervisor: Gregory W. Wornell
Title: Associate Professor, Department of EECS, M.I.T.

Thesis Supervisor: Richard C. Rose
Title: Principal Member of Technical Staff, AT&T Labs – Research

Acknowledgments:

This thesis work was performed as part of the MIT EECS VI-A internship program. The experimental study in this thesis was completed at *AT&T Labs – Research* located at Florham Park, NJ, in the Laboratory of Speech & Image Processing Service Research, under the supervision of Dr. Richard C. Rose.

I would like to first thank Rick for being a greater mentor and a good friend. He has given me valuable guidance over the nine months I spent at AT&T and through the entire process of writing this thesis. During my stay at AT&T, he guided me through understanding the theory and developing the experiments for this thesis. He was always available when I needed help. He not only helped me a great deal at work, he also did me a big favor by driving me to and from work everyday for the last three months of my internship, which was really beyond his obligation as a mentor. During the past three months, Rick took on the responsibility of being the editor of this thesis. He edited each chapter carefully and discussed his commands with me one by one. I have learned a lot from him through this process. Rick has put in tremendous amount of time and effort into this thesis. Without him this thesis would not have been possible.

I would also like to thank other people at AT&T Labs who have helped me during my stay at AT&T. They showed me different aspects of research, broadened my horizon and enriched my internship experience. In particular, I would like to thank Alex Potamianos, Vincent Goffin, Cecilia Castollo, Jerry Wright, and Beppe Riccardi.

I would like to thank my friends for their support over this past year. I would like to thank Alice Wang for putting up with being my roommate during our internship and for being a fun person to spend time with.

Finally, I would like to thank my family for their unconditional love. With the hope to create better opportunities for me, my parents moved our family to the United States, leaving all that they have established behind. To them, I owe my deepest gratitude. To my brother, who is coming to America in less than a month, I give him the best of wishes.

Table of Contents

1. Introduction

Over the last several decades, advances in automatic speech recognition (ASR) research and advances in computing technology have resulted in ASR systems that are capable of handling highly complex spoken language tasks. Current research has emphasized the development of systems that can accept naturally spoken utterances and that are robust against speaker and environmental variability. Speech recognition technology has also extended beyond research laboratories and stepped into the lives of the general public through applications in areas such as telecommunications and education. Despite the advances in speech recognition technology, in the presence of ill-formed utterances and unexplained corrupting influences, many systems still fail to perform recognition accurately. Therefore, it is necessary to have a mechanism for effectively verifying the accuracy of portions of the recognition hypothesis. The goal of this thesis is to investigate the potential role of *utterance verification* (UV) procedures as a means for dealing with these failure modes for large vocabulary *continuous speech recognition* (CSR) and *spoken language understanding* (SLU) systems.

This chapter first describes these issues in more detail to motivate the need for utterance verification, and then proposes a means for its implementation. At the end of this chapter, an outline for the thesis is provided.

1.1 Problem Description

Utterance verification is motivated by several problems that arise in speech recognition and spoken language understanding tasks designed to be used by untrained users in unpredictable acoustic environments. These problems include out-of-domain speech utterances, signal degradation, and variability associated with spontaneous speech.

The first problem is the tendency for untrained users to speak utterances that are out of the domain for which the speech recognizer was configured. There are many examples of out of domain utterances. First, a sentence could be semantically out-of-domain with respect to the set of tasks the recognizer is trained to handle. Second, a sentence could be syntactically ill-formed, which may imply a sentence structure that was not expected by the system. Third, a sentence could contain out-of-vocabulary words which were not placed in the recognizer's vocabulary during training. In any of the above cases, the sentence is considered to be out of domain. When such a sentence is being decoded, the recognizer could only search through its pre-stored vocabulary and sentence formation rules to form a hypothesized word string that best matches the spoken utterance. Since the sentence is out of domain, unfamiliar to the recognizer, the hypothesized result often turns out incorrect.

The second problem is signal degradation caused by unpredictable acoustic environments and channel distortion. Noisy background in acoustic environments interacts with the speech signal in an additive manner, while channel distortion, such as one associated with a telephone channel, interacts with the speech signal in a convolutional

manner. These signal degradation problems often lead to performance degradation in various kinds of recognition tasks.

The third problem is the variability in spontaneous speech. First, utterances spoken spontaneously tend to have varying speaking rates. Second, spontaneous speech often contains disfluencies such as false starts and filled pauses (e.g., *uh*). It is often difficult to establish models for these sources of variability. For example, it is very difficult to automatically detect where any disfluency has occurred in the middle of a sentence. As a result, these disfluencies may be interpreted by the recognizer as vocabulary words.

The above problems of out of domain utterances, signal degradation, and variability in spontaneous speech often result in many recognition errors. In a spoken language understanding system, where utterances are interpreted, these recognition errors often cause misinterpretation of the utterances, which sometimes leads to the SLU commanding wrong actions to be taken.

This thesis develops an utterance verification technique to effectively verify the accuracy of portions of the recognition hypothesis by assigning confidence measures to each decoded word. Being able to identify recognition errors is the first step towards reducing the consequences of the errors. Error detection could provide extra information to the interpretation process in the SLU system. Instead of blindly assuming all words are decoded with equal confidence, the SLU system can incorporate knowledge of word level confidence measures provided by UV in the process of interpreting the utterance, which may lead to improvement in the overall SLU performance.

1.2 Proposed Solution

The goal of this thesis is to develop an utterance verification produce which can determine whether each word in the recognition hypothesis is correct or incorrect. UV is often considered as a hypothesis testing problem. In this thesis, we will investigate the potential of a UV procedure based on a *likelihood ratio* (LR) criterion, which is often used in hypothesis testing. In this section, we introduce the notion of LR based UV and discuss the issue of identifying the acoustic models necessary for performing such test.

This thesis implements a two-pass UV procedure. First, an input utterance is passed through a continuous speech recognizer to produce a hypothesized word string. The resulting word string together with the speech sequence are then passed through the UV unit in which confidence measures are computed for each decoded word. Each word level confidence measure is then compared to a threshold to determine whether to accept or reject the hypothesis that the given word was correctly decoded. The confidence measures can also be used by the SLU system as additional information for interpreting the utterance.

When considering UV as a hypothesis testing problem, the event of a decoded word being correctly decoded corresponds to the null hypothesis, and the event of a decoded word being incorrectly decoded corresponds to the alternative hypothesis. In a LR based UV procedure, the confidence measure assigned to each hypothesized word is calculated from the ratio of the likelihood of the word being correctly decoded with respect to the likelihood of the word being incorrectly decoded. To estimate these two likelihoods, two sets of probabilistic models are used, the null hypothesis, or "target" models, and the

alternative hypothesis models. It is assumed that correctly decoded words are acoustically modeled by target models and incorrectly decoded words are acoustically modeled by alternative models. To calculate a confidence measure for a segment of speech, the speech segment is first compared against each set of models separately, yielding the two likelihoods. The ratio is then taken and converted to a confidence measure. In the LR based UV procedure investigated in this thesis, it is this likelihood ratio based confidence measure that is assigned to each word.

In this thesis, we will also investigate the training of the target and alternative models, which are needed for the UV procedure. The commonly used procedure for training acoustic models in most speech recognition tasks is the maximum likelihood (ML) training procedure. Using a ML criterion to train the target and alternative models may yield reasonably good UV performance. However, it is not directly related to the LR criterion that is used in UV. In this thesis, we investigate a discriminative training procedure based on a gradient decent algorithm. The goal is to increase the separation in the likelihood ratios obtained for correctly and incorrectly decoded words. In this algorithm, a cost function that is related to this separation is defined. Target and alternative model parameters are then re-estimated to optimize this cost function.

In this thesis, the testing and training procedures described above are implemented and evaluated. The measure used to evaluate the UV performance is the percentage error for classification of the hypothesized words being correct or incorrect, which are derived from the confidence measure distributions of the two classes. We experimented with using both ML and discriminative training procedures to train the target and alternative models.

1.3 Thesis Outline

The body of the thesis includes background materials, a description of the theoretical development of the UV algorithms, a description of the experimental study performed to evaluate these algorithms, and conclusions. There are six major chapters:

Chapter 2 provides some of the background knowledge necessary for further discussion of the LR based UV algorithm. First, hidden Markov model (HMM) based maximum likelihood CSR procedure is outlined, introducing notations that will be used throughout this thesis. Next, a review of previous works related to UV is presented.

Chapter 3 presents the theory behind the LR based UV algorithm. First, the testing algorithm is described, providing formulations for calculating the LR based confidence measures. Next, the discriminative training procedure based on the gradient decent algorithm for training both the target and alternative acoustic models is presented. Last, a brief discussion is given on the integration of the utterance verification procedures with statistical language modeling and spoken language understanding.

The next three chapters are dedicated to the description of the experiments and the discussion of the results. In Chapter 4, first, the speech recognition and understanding task, the speech corpus, and the recognition model and performance are described. Next, testing and training procedures are outlined followed by a description of the baseline experiments and a discussion of their results. In the baseline experiments, the target and alternative models are trained using the ML training procedure. These baseline experiments are considered as phase I of the experimental study.

Chapter 5 covers phase II of the experimental study, which employs the usage of discriminative training. The modified training procedure is first described, followed by the presentation and discussion of the experimental results. Next, the rate of convergence of the training procedure is examined. At the end, a discussion of various issues related to the training and testing procedures for UV is given.

Chapter 6 describes three additional experiments related to further applications of acoustic confidence measures. The first experiment investigates using sentence level confidence measures for rejecting utterances that contain only background, noise, silence or non-speech utterances. The second experiment investigates using phrase level confidence measures for utterance verification and compares the result to word level UV. The last experiment implements a method for converting LR based confidence measures to *a posterior* probabilities of each word being correctly decoded given its confidence measure.

Chapter 7 concludes this thesis with a summary and a discussion of possible future work. It is important to review what we have learned from this thesis and to look ahead for where its result may lead us to.

2. Background

This chapter provides some of the background knowledge for further discussion of the likelihood ratio based confidence measure calculation and model training procedures that will be described in detail in Chapter 3. First, hidden Markov model (HMM) based maximum likelihood (ML) continuous speech recognizer (CSR) procedure is outlined. The structure of a typical HMM model is described and notation that will be used throughout this thesis are introduced. Most of the material covered in Section 2.1 and more thorough descriptions of HMM based CSR technology can be found in tutorial references (e.g., [1]). The next three sections are dedicated to a review of previous work on topics including HMM based keyword spotting, various techniques for computing confidence measures, and discriminative training procedures for adjusting model parameters to optimize verification performance.

2.1 HMM Based CSR

A typical HMM based continuous speech recognition system generally consists of three major components. The first component is the front-end analysis which reduces a sampled speech waveform to a sequence of feature vectors, $\mathbf{Y} = \mathbf{y}_1, \cdots, \mathbf{y}_T$. Each feature vector, $\mathbf{y}_t$, is computed from a 10-30 *ms* interval of speech over which the speech signal is assumed to be approximately short-time stationary. The feature vectors provide a representation of the smoothed spectral envelope of the speech. In this work, the features

actually correspond to the cepstrum which is obtained from a linear transformation of the log of a non-uniform filter bank as described in [1a].

The second component of HMM based CSR is the acoustic match between the feature vectors and the acoustic models which are in the form of hidden Markov models, which will be described in Section 2.1.1. The third component of HMM based CSR is the search of optimum word string under the constraint of a statistical language model, which will be described in Section 2.1.2.

2.1.1 Hidden Markov Models

In continuous speech recognition applications, the vocabulary size can be anywhere from several words to 100,000 words. Since it would be impractical to build one statistical model for each word, subword acoustic units are defined so that each word, w, is represented by a concatenation of subword acoustic units, u_l, i.e., $w = u_1 u_2 \cdots u_L$. The rule for this composition is described in the lexicon. There are many possible definitions of these subword units [1b]. A unit can correspond to a phoneme, a syllable, or some other unit. The set of subword units used in this work will be described in Chapter 4. In this work, one HMM model is trained for each subword unit and is denoted by λ_u. The set of all HMM models for all subword units is denoted by Λ.

The underlying goal of HMM is to statistically model both the values of the features vectors and their evolution in time. Each HMM model consists of two major components — a discrete Markov chain with J states, $s_1, \cdots, s_J$, and observation feature vector distributions associated with each state, $b_{s_j}(\mathbf{y})$ $j = 1, \cdots, J$. The Markov chain is described by a set of transition probabilities, a_{ij} $i, j = 1, \ldots, J$, and a set of initial state

probabilities, $\pi_j = a_{0j}$, The feature vector distribution, $b_{s_j}(\mathbf{y}_t) = P(\mathbf{y}_t | q_t = s_j)$, is the probability of emitting feature vector $\mathbf{y}_t$ when occupying state s_j at time t. Here, we use q_t to denote the state association at time t. The feature vector distributions can be modeled in many ways. In our work, each feature vector distribution is modeled as a continuous mixture of M Gaussian pdfs:

$$b_{s_j}(\mathbf{y}) = \sum_{m=1}^{M} c_{jm} N(\mathbf{y}; \mu_{jm}, \Sigma_{jm}). \tag{2.1}$$

In Equation (2.1), c_{jm} denotes the mixture weight of the mth Gaussian pdf of state s_j. The mixture weights of each state must sum to unity, i.e., $\sum_{m=1}^{M} c_{jm} = 1$. The mth Gaussian pdf is represented as $N(\mathbf{y}; \mu_{jm}, \Sigma_{jm})$ with mean vector μ_{jm} and covariance matrix Σ_{jm}, which is assumed to be diagonal in our work. In summary, the set of parameters, $\{\pi_j, a_{ij}, c_{jm}, \mu_{jm}, \Sigma_{jm}\}$, $i, j = 1, \ldots, J$, $m = 1, \ldots, M$, defines a J state, M mixture HMM model. In this work, all acoustic models used are HMM models.

2.1.2 Maximum Likelihood Recognition Method

In maximum likelihood decoding, the goal is to find the most probable word string given the acoustic observations. The objective is to maximize $P(W|\mathbf{Y})$, where $W = w_1, \cdots, w_K$ denotes a word string, and $\mathbf{Y} = \mathbf{y}_1, \cdots, \mathbf{y}_T$ denotes the observed sequence of feature vectors. Using Baye's rule, this criterion can be rewritten as

$$\arg\max_W P(W|\mathbf{Y}) = \arg\max_W P(\mathbf{Y}|W) P(W). \tag{2.2}$$

In Equation (2.2), *P(W)* represents the match with the statistical language model and *P(***Y***|W)* represents the match with the acoustic HMM models.

The acoustic match for each word, $P(\mathbf{Y}_k|w_k)$ is estimated by matching the sequence of feature vectors corresponding to the k[th] word, $\mathbf{Y}_k$, with the HMM models of the sequence of subwords contained in that word. If we assume the Markov chain occupies state s_j at time t, i.e., $q_t = s_j$, then according to the definition of HMM model,

$$P(\mathbf{Y}_k, Q|w_k) = P(\mathbf{y}_1,\cdots,\mathbf{y}_T, q_1,\cdots,q_T|\Lambda_k) = \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{y}_t), \qquad (2.3)$$

where $Q = q_1,\cdots,q_T$ is the state sequence information. It is often referred to as *segmentation* and is in most cases hidden or can not be obtained directly. In order to obtain a reasonable estimation of the segmentation, the Viterbi algorithm [1c] is used to search through all possible state sequences to find the one that would yield the maximum value for the likelihood function, that is,

$$Q^* = \arg\max_{Q} P(\mathbf{Y}_k, Q|w_k) = \arg\max_{q_1\cdots q_T} \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{y}_t). \qquad (2.4)$$

Given the segmentation Q^*, the probability $P(\mathbf{Y}_k|w_k)$ can then be approximated as $P(\mathbf{Y}_k, Q^*|w_k)$.

The second factor in Equation (2.2), $P(W) = P(w_1, w_2,\cdots,w_K)$, is estimated from the language model, which provides a probabilistic description of the syntax of the language. In an n-gram language model, the sequence of words, *W*, is assumed to be Markovian, so that

$$P(w_k|w_{k-1},\cdots,w_1) = P(w_k|w_{k-1},\cdots,w_{k-n+1}). \qquad (2.5)$$

The order of the model, n, is typically limited to two (bigram) or three (trigram). For example, in the bigram case, the probability of a word string can be expressed as

$$P(W) = P(w_1, w_2, \cdots, w_k) = P(w_1) \cdot P(w_2|w_1) \cdots P(w_K|w_{K-1}) . \qquad (2.6)$$

The language model complexity is measured by perplexity, which is the exponential of the average word entropy at a decision point in the grammar, $2^{H(w)}$ [1d].

The problem of training or estimating the parameters of the HMM models from speech data will not be discussed here in detail. The goal in speech recognition is to determine the word sequence W that maximizes $P(W|\mathbf{Y}, \Lambda)$. In the ML training procedure, the word sequence W is known, and the goal is to determine the model parameters, $\Lambda = \{\pi_j, a_{ij}, c_{jm}, \mu_{jm}, \Sigma_{jm}\}$, which maximize $P(\mathbf{Y}|W, \Lambda)$. The reader is referred to published tutorial references for discussion of the Baum-Welch algorithm and the segmental k-means algorithm which are used for HMM model training [1-2].

2.2 Keyword Spotting

Many speech recognition systems rely on extracting partial information from unconstrained speech utterances. These utterances could be ill-formed and contain out-of-vocabulary (OOV) words. One approach which has been studied intensively is keyword spotting, which involves detecting the occurrence of a given set of information bearing words in running speech [3-8]. In an HMM based keyword spotting system, HMM acoustic models are trained for each keyword, and a set of "filler" or "background" models are also trained to represent OOV words or non-keyword speech. CSR techniques are applied to search through the network of keyword and filler models to produce a

continuous stream of decoded keywords and fillers. One major element that distinguishes a keyword spotting task from a CSR task is that keyword spotting only calls for the detection of a relatively small subset of the total words that might appear in the input utterances. Some of the more recent research in keyword spotting has demonstrated that more detailed modeling of the non-keyword speech can result in significant improvement in word spotting performance [6-8].

In most keyword spotting systems, a second stage decision rule is used to verify each keyword occurrence hypothesized by the CSR. A score or confidence measure is calculated and then compared to a preset threshold to determine whether to accept or reject a hypothesis. Tradeoffs between probability of detection and false alarm rate can be achieved by adjusting this threshold. Some of these keyword spotting systems operate in a mode which is similar to the utterance verification system being investigated in this thesis except only the events of keyword occurrences are being verified.

In our work, a large vocabulary CSR (LVCSR) system with subword HMM models is used. Unlike keyword spotting systems, we attempt to obtain a complete transcription of the input utterance. All words are treated equally in decoding and are all assigned confidence measures in verification. It is not until all decoded words and their confidence measures are passed to the spoken language understanding unit that individual words or phrases might be treated as containing significant content

2.3 Confidence Measures

In many speech recognition systems, confidence measures have been used for verification of recognition hypotheses. In the keyword spotting systems discussed above,

confidence measures are used for acceptance and rejection of hypothesized keyword occurrences. In other tasks where a large percentage of the input utterances are out of domain, confidence measures can be used to detect illegitimate words and utterances [9,11,15,16,18]. In tasks where OOV words and ill-formed utterances are rare, confidence measures can be used to detect events where in-vocabulary words are decoded incorrectly [13].

There are many different techniques for computing confidence measures [6,8-12,14,16,18]. In [6], log-likelihood scores obtained directly from Viterbi decoding are used as keyword scores. In many recent systems, various forms of likelihood ratio scores have been used as confidence measures, and have been shown to out-perform likelihood scores. LR scores have less variability, and are more efficient and robust than likelihood scores.

As with any hypothesis test, the LR approach involves distinguishing a null hypothesis from an alternative hypothesis. There have been many different ways for defining the distributions associated with these hypotheses and for estimating their likelihoods in the context of speech recognition. This is especially true for the alternative hypothesis. One approach is to estimate the likelihood associated with the alternative hypothesis by summing over all the likelihoods associated with other hypotheses that are possible [10-12]. For example, one system for verifying the presence of individual keywords in a continuous utterance generates an N-best list of 500 hypothesized sentences using a LVCSR system [10]. The null hypothesis for testing for the presence of a keyword is formed from all of the hypothesized sentences that contain the keyword. The alternative hypothesis is formed from all of the hypothesized sentences that do not contain the

keyword. The likelihood of each hypothesis is estimated by summing over the likelihoods of all hypothesized sentences that are associated with that hypothesis.

Another approach that has been used to estimate LR based confidence measures is to train designated acoustic models to represent the alternative hypothesis, which is defined to be the event of a spotted keyword being a false alarm or a decoded word being incorrect. Likelihoods associated with the alternative hypothesis are calculated by matching speech segments against the alternative acoustic models. In some keyword spotting systems, background and filler models are used as alternative acoustic models [3,14]. In other systems, designated anti-keyword or "impostor" models are used [13,16,18]. In this work, designated subword acoustic models are trained for the purpose of representing the alternative hypothesis in the likelihood ratio calculation.

2.4 Discriminative Training

In the LR based confidence measure calculation described above, two sets of acoustic models are needed, the "target" models for the null hypothesis, and the alternative hypothesis models. Various methods for constructing these models and for discriminatively training their parameters have been investigated [13-18]. The goal of discriminative training is to maximize the discrimination power of the likelihood ratio test. It has been shown to out-perform the ML training method for many keyword spotting and utterance verification tasks. Different forms of models have been investigated. These include alternative hypothesis models that are word level single state HMM's [16]. In this work, more sophisticated impostor models are used, and discriminative training methods are used for re-estimating both target and impostor models [18, 20].

2.5 Summary

This chapter gave some of the background knowledge that is necessary for understanding the verification and training algorithms that will be described in the next chapter. First, the HMM based CSR procedure using ML criterion is described. The structure of a typical HMM model is described and notation is introduced. Each HMM model is characterized by a discrete Markov chain, $\{\pi_j, a_{ij}\}$ and feature vector distributions, which are continuous mixtures of Gaussian pdfs, $b_{s_j}(\mathbf{y}) = \sum_{m=1}^{M} c_{jm} N(\mathbf{y}; \mu_{jm}, \Sigma_{jm})$. The ML recognition procedure is described as a technique for finding the most probable word string given the acoustic observations based on pre-trained acoustic and language models, i.e., to find $\arg\max_W P(W|\mathbf{Y})$ by maximizing $P(\mathbf{Y}|W)P(W)$.

Previous work on topics of HMM based keyword spotting, confidence measures computation, and discriminative training are reviewed. This work investigates confidence measure calculation and discriminative model training algorithms similar to some that have been recently studied. The significance of this work is to apply those techniques to a more difficult task which deals with highly unconstrained speech, which will be described in Chapter 4.

Another novelty of this work is that it lays the ground work for future studies of the potential for integrating utterance verification procedures with language modeling and spoken language understanding. There has been little work done in this field, and we hope that this study could lead to further applications.

3. Confidence Measures and Training Algorithms for Utterance Verification

This chapter provides a detailed description of likelihood ratio based confidence measures used for utterance verification and of the algorithm used for training the acoustic models used for UV. In the first section of the chapter, the implementation of the LR based UV procedure in continuous speech recognition is described, providing detailed formulations for calculating the LR based confidence measures. In the second section, a discriminative training procedure based on a gradient descent algorithm is presented as a means for training both the target and alternative acoustic models. The training procedure is designed to optimize a LR criterion which is very similar to that used in verification. In the last section there is a brief discussion relating to the integration of the UV procedure with statistical language modeling and the spoken language understanding system.

3.1 Testing Procedure and Confidence Measure Calculation

The likelihood ratio based utterance verification system being investigated in this work consists of two passes. First, an input utterance is passed through a continuous speech recognizer (CSR). Second, the resulting decoded word string together with the observed feature vector sequence are passed through the UV unit in which a confidence measure is computed for each word. This confidence measure can then be compared to a threshold to determine whether a word is correctly decoded.

Utterance verification is often considered as a hypothesis testing problem. The event where a word is correctly decoded is defined to be the null hypothesis, C=1; and the event where a word is incorrectly decoded is defined as the alternative hypothesis, C=0. The likelihood ratio is then the ratio of the *a posteriori* probabilities of observing a feature vector sequence **Y** conditioned on the two events,

$$\frac{P(\mathbf{Y}|C=1)}{P(\mathbf{Y}|C=0)} \quad . \tag{3.1}$$

It is assumed that for a correctly decoded word, the sequence of observation vectors, **Y**, is modeled by a set of null or target hypothesis models, Λ^c, and for an incorrectly decoded word, **Y** is modeled by a set of alternative hypothesis models, Λ^a. The likelihood ratio equation then becomes

$$\frac{P(\mathbf{Y}|\Lambda^c)}{P(\mathbf{Y}|\Lambda^a)} \quad . \tag{3.2}$$

In this work, both Λ^c and Λ^a are hidden Markov models. Furthermore, it is assumed that for each subword unit, u, there are dedicated HMMs λ_u^c and λ_u^a. The target hypothesis model, λ_u^c, is similar to and could be identical to the HMM model used in the CSR decoder. The alternative hypothesis model, λ_u^a, models incorrectly decoded words. It is taken to be a combination of a background model, λ^{bg}, shared by all subword units and a set of subword unit specific impostor models, λ_u^{im}. The purpose of the background model is to provide a representation of the generic spectral characteristic of speech. The purpose of the impostor models is to provide a representation of acoustic events that are

frequently confused with a given subword unit. The background and impostor model probabilities are combined linearly in the following manner,

$$P(\mathbf{y}|\lambda^{a}) = (1-\alpha)\cdot P(\mathbf{y}|\lambda^{im}) + \alpha \cdot P(\mathbf{y}|\lambda^{bg}). \tag{3.3}$$

where α is a weighting constant. For simplicity, in this thesis, we will use the model topology that each pair of target model and impostor model for one unit contain the same number of states and the background model is a single state model.

The confidence measures are calculated from the observed feature vector sequence, **Y**, the decoded word sequence, *W*, and the segmentation information on the start and end time of each state. In Section 2.1.2 it was mentioned that given an HMM model, Λ, and a sequence of observation vectors, **Y**, a sequence of states, $Q^{*} = q_1,\cdots,q_T$, can be obtained such that $P(\mathbf{Y},Q^{*}|\Lambda) = \max_{Q} P(\mathbf{Y},Q|\Lambda)$ where Q is any possible sequence of states. Hence, given this state sequence, Q^{*}, the probability $P(\mathbf{Y}|\Lambda)$ can be approximated as

$$P(\mathbf{Y},Q^{*}|\Lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} b_{q_t}(\mathbf{y}_t). \tag{3.4}$$

As a result the likelihood ratio in Equation (3.2) can be approximated as

$$\frac{P(\mathbf{Y},Q^{*}|\Lambda^{c})}{P(\mathbf{Y},Q^{*}|\Lambda^{a})} = \frac{\prod_{t=1}^{T} a^{c}_{q_{t-1}q_t} b^{c}_{q_t}(\mathbf{y}_t)}{\prod_{t=1}^{T} a^{a}_{q_{t-1}q_t} b^{a}_{q_t}(\mathbf{y}_t)}. \tag{3.5}$$

It is assumed in Equation (3.5) that the state sequence Q^{*} is the same in both the null hypothesis and the alternative hypothesis and was obtained using the Viterbi algorithm to maximize $P(Y,Q|\Lambda^{c})$. If the transition probabilities in Equation (3.5) are also assumed to

be equal, i.e., $a_{ij}^c = a_{ij}^a$, then the approximation to the logarithm of the likelihood ratio can be written as

$$\begin{aligned} R &= \log \frac{P(\mathbf{Y}, Q^* | \Lambda^c)}{P(\mathbf{Y}, Q^* | \Lambda^a)} = \log \prod_{t=1}^{T} \left(\frac{b_{q_t}^c(\mathbf{y}_t)}{b_{q_t}^a(\mathbf{y}_t)} \right) \\ &= \sum_{t=1}^{T} \left(\log b_{q_t}^c(\mathbf{y}_t) - \log b_{q_t}^a(\mathbf{y}_t) \right), \end{aligned} \tag{3.6}$$

where $$b_{q_t}^a(\mathbf{y}_t) = \alpha \cdot b^{bg}(\mathbf{y}_t) + (1-\alpha) \cdot b_{q_t}^{im}(\mathbf{y}_t).$$

This is the general form of the log-likelihood ratio, or *LR score R*, computed over a speech segment consisting of T observation frames.

The calculation of word level LR based confidence measures consists of a hierarchy of calculations of LR scores at frame level, state level, unit level and finally word level. At the frame level, the LR score for frame t with observation vector $\mathbf{y}_t$ and state q_t takes the form,

$$\begin{aligned} R_t(\mathbf{y}_t) &= \log b_{q_t}^c(\mathbf{y}_t) - \log b_{q_t}^a(\mathbf{y}_t) \\ &= \log b_{q_t}^c(\mathbf{y}_t) - \log \left[\alpha \cdot b^{bg}(\mathbf{y}_t) + (1-\alpha) \cdot b_{q_t}^{im}(\mathbf{y}_t) \right]. \end{aligned} \tag{3.7}$$

At the state level, the LR score associated with the segment of speech over which the frames were decoded as "belonging" to state s_j is taken to be the sum of the LR scores of all frames in the state, normalized by the state duration, T_j,

$$R_{s_j}(\mathbf{Y}_{s_j}) = \frac{1}{T_j} \sum_{t=ti_j}^{tf_j} R_t(\mathbf{y}_t) \ . \tag{3.8}$$

In Equation (3.8), ti_j, and tf_j are the time indices of the first and last frames, such that $q_t = s_j$ for $ti_j \le t \le tf_j$, $T_j = tf_j - ti_j + 1$, and $\mathbf{Y}_{s_j} = \mathbf{y}_{ti_j}, \cdots, \mathbf{y}_{tf_j}$. The reason for doing the normalization is so that different states with different durations would have comparable scores. Finally, the unit level LR score is the average of the state level scores,

$$R_u(\mathbf{Y}_u) = \frac{1}{J_u} \sum_{j=1}^{J_u} R_{s_j}(\mathbf{Y}_{s_j}), \tag{3.9}$$

where J_u is the number of states in the HMM model for unit u and $\mathbf{Y}_u = \mathbf{Y}_{s_1}, \cdots, \mathbf{Y}_{s_{J_u}}$.

The word level confidence measure R_w is formed from weighted linear combinations of unit level LR scores, $R_u(\mathbf{Y}_u)$. As with any likelihood ratio based measures, $R_u(\mathbf{Y}_u)$ can exhibit a wide dynamic range. In order to reduce the effects of this dynamic range, a continuous nonlinear transformation, $F_u(R_u(\mathbf{Y}_u))$, is applied to the unit level scores. The nonlinear transformation is derived from the well known sigmoid function,

$$F_u(\mathbf{Y}_u) = \frac{1}{1 + \exp(-\gamma \cdot (R_u(\mathbf{Y}_u) - \tau))}, \tag{3.10}$$

where γ and τ are constants chosen for the function. The offset value τ determines the center of the weighting and the scale factor γ is related to the width of the function. This function maps the entire real axis to the unit interval, (0,1). A sample of a sigmoid function with $\tau = 0$ and $\gamma = 1$, i.e. $y = \frac{1}{1 + \exp(-x)}$ is plotted in Figure 3-1.

The word level score is the geometric mean of the weighted unit level LR score scores, that is

$$R_w(\mathbf{Y}_w) = \exp\left(\frac{1}{N_w}\sum_{i=1}^{N_w}\log\left(F_{u_i}(\mathbf{Y}_{u_i})\right)\right), \tag{3.11}$$

where N_w is the number of subwords contained in word w, and i is an index used for summation over these subwords. The effect of the geometric mean is to assign greater weights to units with low LR scores, which are more likely to have been incorrectly decoded. As a result, an individual unit with a particularly low score can cause the word level score to be low and consequently cause the word to be classified as a false alarm and be rejected. This reflects the rule of considering a decoded word to be incorrect when any one of its subword unit is misdecoded.

In our work, the final score, R_w, is defined as the confidence measure for a word. Comparing it to a threshold yields the decision of whether to accept or reject the hypothesis that the word had been correctly decoded. Words with confidence measure

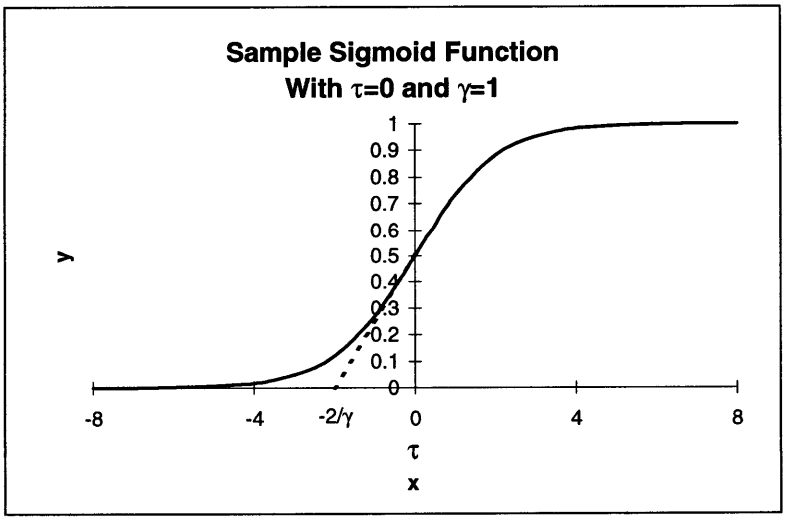

Figure 3-1 Sample sigmoid function with $\tau = 0$ and $\gamma = 1$, that is $y = \frac{1}{1+\exp(-x)}$. The center of the function is determined by τ, where the slope is $\gamma/4$.

above the threshold are accepted and words with confidence measure below the threshold are rejected, namely,

$$R_w \underset{reject}{\overset{accept}{\gtrless}} \xi . \tag{3.12}$$

It is worth mentioning that the likelihood, $P(\mathbf{Y}|W)P(W)$, obtained from maximum likelihood decoding cannot be directly used as a confidence measure. The reason is the following. Although it is true that

$$\arg\max_W P(W|\mathbf{Y}) = \arg\max_W P(\mathbf{Y}|W)P(W),$$

the decoder itself does not produce an estimate of $P(W|\mathbf{Y})$,

$$P(W|\mathbf{Y}) = \frac{P(\mathbf{Y}|W)P(W)}{P(\mathbf{Y})} \neq P(\mathbf{Y}|W)P(W). \tag{3.13}$$

This is to say that, although $P(\mathbf{Y}|W)P(W)$ is sufficient for determining the best word sequence, it is not the same as the likelihood of a word sequence being correct, because it is not normalized by the probability of observing the sequence, $P(\mathbf{Y})$. This means that in the ML procedure, a decoded word with higher likelihood score is not necessary more likely to be correct than another decoded word with lower likelihood score when they correspond to different segments of speech. This is the reason why likelihood scores from the ML recognition process cannot be used directly to verify whether a word is correctly decoded. The advantage of using the likelihood ratio algorithm proposed above is that the probability of observing the sequence, $P(\mathbf{Y})$, gets canceled out as the ratio is taken, thus normalization is no longer needed.

3.2 Discriminative Training Algorithm for LR Based UV

This section describes an algorithm for training the model parameters associated with the likelihood ratio based confidence measure calculation that was presented in Section 3.1. The probability distributions that represent the target and alternative hypotheses in the LR based hypothesis testing were both parameterized as hidden Markov models. The goal of the training procedure is to re-estimate the model parameters to optimize a cost function that is directly related to the LR criterion used in verification as defined in Section 3.1.

This section has three parts. First, the cost function is defined and motivated in terms of the objective of optimizing UV performance. Second, a gradient descent algorithm for minimizing this cost function is described. Finally, the parameter update equations are obtained by solving the partial derivatives associated with the gradient descent algorithm.

3.2.1 Cost Function Definition

The goal in defining a cost function for UV is to assign low cost to "desirable" events, and to assign high cost to "undesirable" events. The terms "desirable" and "undesirable" in this context refer to the ability of the unit level LR score $R_u(\mathbf{Y}_u)$ to predict whether a unit has been correctly decoded. Hence, desirable event would correspond either to a unit being correctly decoded and $R_u(\mathbf{Y}_u)$ being very large, or to a unit being incorrectly decoded and $R_u(\mathbf{Y}_u)$ being very small. Other events corresponding to LR scores not predicting the accuracy of the hypothesized units are considered undesirable. Table 3-1 describes the relative costs in terms of the possible events.

Objective For Cost Function Definition

hypothesized unit	unit level likelihood ratio score	
	low	high
correctly decoded	$$$	$
incorrectly decoded	$	$$$

Table 3-1 The dark shaded entries correspond to undesirable events, which should be assigned high cost. The light shaded entries correspond to desirable events, which should be assigned low cost.

A cost function which reflects the above objective and is also continuous and differentiable is derived from the sigmoid function,

$$F_u\left(R_u(\mathbf{Y}_u)\right) = F_u\left(\mathbf{Y}_u, \lambda_u^c, \lambda_u^{im}\right) = \frac{1}{1+\exp\left(-\gamma \cdot \delta(u)(R_u(\mathbf{Y}_u) - \tau)\right)}, \tag{3.14}$$

where γ and τ are fixed parameters and the indicator function $\delta(u)$ is defined as

$$\delta(u) = \begin{cases} -1 & u \text{ is correctly decoded} \\ 1 & u \text{ is incorrectly decoded} \end{cases}.$$

The goal is to adjust the parameters of λ_u^c and λ_u^{im} to minimize the expected value of the cost function $F_u\left(\mathbf{Y}_u, \lambda_u^c, \lambda_u^{im}\right)$. The background model, λ^{bg}, which represents the generic spectral characteristic of speech, will not be re-estimated in this work. A plot of the cost function vs. unit level LR score for $\tau = 0$ and $\gamma = 1$ is shown in Figure 3-2.

3.2.2 Gradient Descent Algorithm

The implementation for this discriminative training procedure with cost function defined in Equation (3.14) has four steps. First, training utterances are decoded by a continuous speech recognizer. Second, the hypothesized word strings are compared to the known transcriptions of these training utterances, and each decoded subword unit is

labeled as either correct or incorrect, i.e., $\delta(u) = \pm 1$. Third, the cost is evaluated according to Equation (3.14) with $R_u(\mathbf{Y}_u)$ defined in Equation (3.9). Finally, a gradient update is performed on the average of the cost, which is a close approximation of the expected value of the cost that would be obtained on unseen data, namely,

$$\overline{F}(u,\lambda_u^c,\lambda_u^{im}) = \frac{1}{N_u}\sum_{i=1}^{N_u} F_{u_i}(\mathbf{Y}_{u_i}, \lambda_u^c,\lambda_u^{im}) , \tag{3.15}$$

$$\Lambda_{n+1}^k = \Lambda_n^k - \varepsilon \cdot \nabla \overline{F} , \tag{3.16}$$

where N_u is the total number of occurrences of the unit u in the training data, and i is an index for summing over these unit. ε is the learning rate constant for the gradient update. k refers to target models ($k = c$) or impostor models ($k = im$).

Note that the magnitude of the derivative of the cost function with respect to the LR score attains its maximum at τ and goes to zero for values away from τ, as shown in

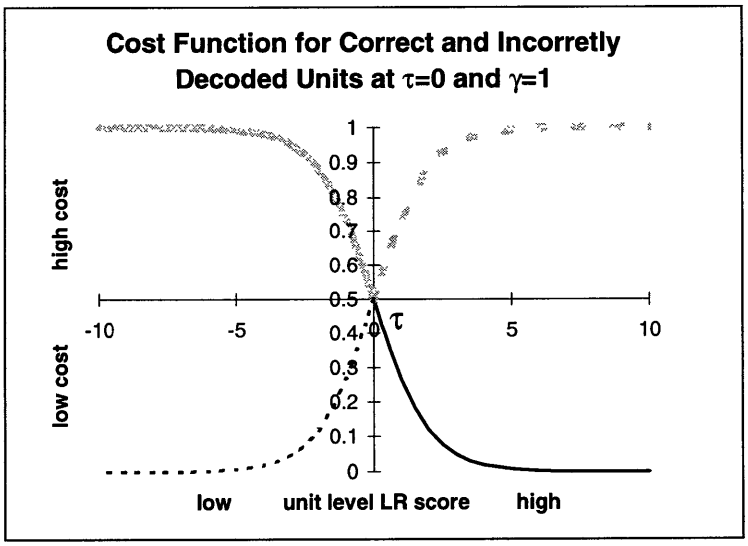

Figure 3-2 Cost function for correctly decoded units is plotted by solid line while the dotted line is for the incorrectly decoded units. Thick gray segments correspond to high cost events and thin black segments correspond to low cost events.

Figure 3-2. As a result, the LR scores close to τ will contribute the most to the gradient computation and consequently would be most affected by the update. In other words, the sigmoid form of the cost function results in a situation where most of the effect occurs near the center where most confusions occur and a small change in LR score could have significant contribution to the prediction of whether a unit is correctly decoded or not. For extreme values of the LR score, changing their value by a small amount would have little impact. Figure 3-2 also implies that by decreasing the cost function using the gradient descent algorithm, the LR scores for the correctly decoded units would tend to shift right, and the LR scores for the incorrectly decoded units would shift left. This would consequently increase the separation between LR scores for correctly decoded and incorrectly decoded units, which is exactly the desired behavior.

3.2.3 Parameter Update Equations

The purpose of this section is to solve for the gradient of the average cost function, $\nabla\overline{F}$, in the gradient update Equation (3.16). Expressions will be obtained for the partial derivatives with respect to the parameters of both the target model, λ_u^c, and the impostor model, λ_u^{im}, of every subword unit u. Hence, the procedure involves simultaneous re-estimation of both target and impostor model parameters.

Recall the procedure for computing the average cost for a unit u, $\overline{F}(u,\lambda_u^c,\lambda_u^{im})$. First, the frame, state, and unit level LR scores are computed for each unit according to Equations (3.7)–(3.9). Next, each unit level LR score is converted to a cost according to Equation (3.14). Finally, the averaged cost for each unit is computed from the costs of all the occurrences of the unit. The gradient of the average cost with respect to each HMM

parameter can then be obtained by repeatedly using the chain rule for derivative computation. For each HMM model, λ_u^k $(k = c, im)$, the parameters to be updated include the mixture weights for state s_j and mixture m, c_{jm}, the mean vectors, $\mu_{jm} = \{\mu_{jmi}\}$, and the diagonal covariance matrix, $\Sigma_{jm} = \{\sigma_{jmi}^2\}$, where $j = 1 \cdots J$ $m = 1 \cdots M$, and i denotes the dimensions of the feature vector.

Let ϕ denote any element of the set of HMM model parameters, $\{c_{jm}, \mu_{jmi}, \sigma_{jmi}^2\}$, of all target models and impostor models. By applying the chain rule, the following partial derivatives are obtained,

$$\frac{\partial \overline{F}(u, \lambda_u^c, \lambda_u^{im})}{\partial \phi} = \frac{1}{N_u} \sum_{i=1}^{N_u} \frac{\partial F_{u_i}}{\partial \phi}, \tag{3.17}$$

$$\frac{\partial F_u}{\partial \phi} = \gamma \cdot \delta(u) \cdot F_u \cdot (1 - F_u) \cdot \frac{\partial R_u}{\partial \phi}, \tag{3.18}$$

$$\frac{\partial R_u}{\partial \phi} = \frac{1}{J_u} \sum_{j=1}^{J_u} \left(\frac{1}{T_j} \sum_{t=ti_j}^{t=tf_j} \frac{\partial R_t}{\partial \phi} \right). \tag{3.19}$$

Recall Equation (3.7) for the definition of frame level LR score, $R_t(\mathbf{y}_t)$,

$$\begin{aligned} R_t(\mathbf{y}_t) &= \log b_{q_t}^c(\mathbf{y}_t) - \log b_{q_t}^a(\mathbf{y}_t) \\ &= \log b_{q_t}^c(\mathbf{y}_t) - \log\left[\alpha \cdot b^{bg}(\mathbf{y}_t) + (1 - \alpha) \cdot b_{q_t}^{im}(\mathbf{y}_t)\right]. \end{aligned} \tag{3.20}$$

The partial derivative against any parameter in a target model, i.e., $\phi^c \in \lambda^c$, is

$$\frac{\partial R_t}{\partial \phi^c} = \frac{\partial \log b_{q_t}^c(\mathbf{y}_t)}{\partial \phi^c}. \tag{3.21}$$

The partial derivative against any parameters in an impostor model, i.e.,$\phi^{im} \in \lambda^{im}$, is

$$\frac{\partial R_t}{\partial \phi^{im}} = \frac{-(1-\alpha)\cdot b_{q_t}^{im}}{(1-\alpha)\cdot b_{q_t}^{im} + \alpha \cdot b^{bg}} \cdot \frac{\partial \log b_{q_t}^{im}(\mathbf{y}_t)}{\partial \phi^{im}}. \tag{3.22}$$

To solve for the partial derivatives $\frac{\partial \log b_{q_t}^{k}(\mathbf{y}_t)}{\partial \phi^{k}}$ $(k = c, im)$, recall the definition of observation probabilities in Section 2.1.1,

$$b_{q_t=s_j}(\mathbf{y}_t) = P\left(\mathbf{y}_t \mid q_t = s_j, \lambda\right), \tag{3.23}$$

where $q_t = s_j$ is the state association at time *t*. When expressed in terms of HMM parameters, the definition of the observation probability is

$$b_{q_t=s_j}(\mathbf{y}_t) = \sum_{m=1}^{M} c_{jm} N(\mathbf{y}; \mu_{jm}, \Sigma_{jm}), \tag{3.24}$$

where $N(\mathbf{y}; \mu_{jm}, \Sigma_{jm})$ represents Gaussian pdfs. By expanding the Gaussian expressions, partial derivatives of $\log b_{q_t=s_j}^{k}(\mathbf{y}_t)$ against each model parameter can be solved.

There are a variety of gradient based HMM model re-estimation procedures in the literature [21,23,24]. In those procedures, the gradients are taken with respect to the transformed parameters, $\bar{c}_{jm}^{k}$, $\bar{\sigma}_{jmi}^{k}$, and the mean vector μ_{jmi}^{k}. The transformations are defined as

$$\begin{aligned} \bar{\sigma}_{jmi}^{k} &= \log \sigma_{jmi}^{k} & \sigma_{jmi}^{k} &= \exp(\bar{\sigma}_{jmi}^{k}) \\ \bar{c}_{jm}^{k} &= \log c_{jm}^{k} & c_{jm}^{k} &= \frac{\exp(\bar{c}_{jm}^{k})}{\sum_{n=1}^{M} \exp(\bar{c}_{jn}^{k})}. \end{aligned} \tag{3.25}$$

There are two main reasons for taking these transforms. The first reason is that it is necessary for the values of the standard deviations and weights to stay positive and for the weights to sum to unity. The transforms above guarantee the preservation of these properties as the parameters are updated. The second reason is that by updating the log of the parameters and then transforming them back, the magnitude of change of each parameter is then scaled by its original value, because $\Delta\phi = \phi \cdot \Delta(\log\phi)$. This way, small values would have small absolute changes and large values have large absolute changes. This is desirable because, for example, a change in standard deviation from 0.001 to 0.101 is much more significant than a change from 10 to 10.1. In the former, the pdf is scaled by a factor of about 100.

Combining Equations (3.24) and (3.25), the gradient with respect to the transformed parameters are

$$\frac{\partial \log b^k_{q_t=s_j}(\mathbf{y}_t)}{\partial \bar{c}^k_{jm}} = \gamma^k_{jm} - c^k_{jm}, \tag{3.26}$$

$$\frac{\partial \log b^k_{q_t=s_j}(\mathbf{y}_t)}{\partial \mu^k_{jmi}} = \gamma^k_{jm} \cdot \frac{(y_{ti} - \mu^k_{jmi})}{\left(\sigma^k_{jmi}\right)^2}, \tag{3.27}$$

$$\frac{\partial \log b^k_{q_t=s_j}(\mathbf{y}_t)}{\partial \bar{\sigma}^k_{jmi}} = \gamma^k_{jm} \cdot \left[\left(\frac{\left(y_{ti} - \mu^k_{jmi}\right)}{\sigma^k_{jmi}} \right)^2 - 1 \right], \tag{3.28}$$

where

$$\gamma^k_{jm} = \frac{c^k_{jm} N(\mathbf{y}_t; \mu^k_{jm}, \Sigma^k_{jm})}{b^k_{q_t=s_j}(\mathbf{y}_t)} \,. \tag{3.29}$$

3.3 Integration of UV With Language Model and SLU

The utterance verification algorithm described in Section 3.1 assigns acoustic confidence measures to each word in the hypothesized word strings produced by a CSR decoder. This section suggests how these acoustic confidence measures can be integrated into an n-gram statistical language modeling framework to facilitate a closer interaction between language and acoustic models. It also suggests how these acoustic confidence measures can be integrated with spoken language understanding in order to facilitate a more accurate semantic interpretation of each utterance.

It is often the case in speech recognition that a word is decoded solely due to the strength of the language model, even when the local acoustic match is poor. Therefore, when a word is incorrectly decoded, the constraints imposed by the n-gram language model often cause the following words to be incorrectly decoded as well. By providing the language model with access to acoustic confidence measures, the effect of this artifact could be reduced. In one study, confidence measures computed from word level acoustic likelihoods were integrated with language modeling by replacing the likelihood score, $P(\mathbf{Y}|W)P(W)$, with $P(\mathbf{Y}|W)^{\alpha}P(W)^{\beta(X)}$, where α is a constant, and β is a function of the confidence measure, X, associated with a word [19]. This corresponds to an ad hoc procedure which weights the language modeling probabilities with acoustic confidence measures. In another study, likelihood ratio based word level acoustic confidence measures are integrated into an n-gram statistical language model so that the language model takes into consideration not only the word history, $w_{k-1},\cdots w_{k-n+1}$, but also a coded representation of the acoustic confidence, $x_{k-1},\cdots,x_{k-n+1}$, associated with the

word history [20]. The conditional language model probabilities take the form $P(w_k|w_{k-1},x_{k-1},\cdots,w_{k-n+1},x_{k-n+1})$ as opposed to the standard n-gram probabilities $P(w_k|w_{k-1},\cdots w_{k-n+1})$. The incorporation of acoustic confidence into the language model results in a general formalism which expands the state space of the language model to directly include acoustic knowledge.

The UV confidence measures described in this thesis were also integrated into the statistical formalism associated with a spoken language understanding system [20]. The task of the SLU system was to classify input utterances according to a set of semantic classes. Probabilities derived from acoustic confidence measures were used to scale the posterior probability of a semantic class given the input utterances. It was found that including acoustic confidence measures resulted in an improvement in SLU performance by allowing for the rejection of semantic class hypotheses with low acoustic confidence measures [20].

3.4 Summary

This chapter has provided a detailed description of the confidence measures used for UV in large vocabulary CSR and has also described a discriminative training procedure for estimating the parameters of UV models. Word level confidence measures are computed as the geometric means of weighted unit level LR scores. The discriminative training procedure is based on a gradient descent algorithm that simultaneously updates both the target and impostor model parameters. The gradient descent algorithm is based on a cost function where decreasing the value of this cost function results in an increase in the separation between the LR scores obtained for correctly and incorrectly decoded subword

units. Detailed formulations for calculating the gradient of the cost function in the HMM model parameter space were presented in Section 3.2. Finally, there is a discussion in Section 3.3 on how the acoustic confidence measures described here might be integrated with statistical language modeling and spoken language understanding.

4. Phase I: Baseline Experiments

The goal of this chapter is to describe a set of baseline experiments for evaluating the performance of the likelihood ratio based utterance verification algorithm described in Section 3.1. The experiments were performed using utterances collected from a large vocabulary spoken language task over the public telephone network. The speech corpus derived from this task will be referred to here as the *"How may I help you?"* (HMIHY) corpus. The results were evaluated as the ability of the UV system to detect correctly decoded vocabulary words in hypothesized word strings produced by a CSR decoder. This chapter has six sections. Section 4.1 describes the HMIHY natural language task, the speech corpus, and the configuration of the baseline CSR system. Section 4.2 describes the testing procedure by providing a flow diagram and a list of the steps involved. Section 4.3 describes the procedure for training initial background, target, and impostor models using a maximum likelihood training algorithm. These models are used later for initialization in the discriminative training procedure which will be described in Chapter 5. Section 4.4 describes each of the three individual experiments, while Section 4.5 presents and discusses the results. Section 4.6 summarizes the chapter.

4.1 Speech Corpus: *How may I help you?*

This section describes the HMIHY speech corpus, which is used to evaluate all algorithms and procedures investigated in this thesis work. This section has three parts. The first part describes how the HMIHY task is structured. The next part describes the

speech corpus, how it was collected, and how it was partitioned into data sets for the experiments in this work. The last part describes the recognition models and provides recognition performance on each of the data sets.

4.1.1 The HMIHY Task

The "*How may I help you?*" (HMIHY) task is an automated call routing service. In this system, a user is first prompted with the open ended question "*This is AT&T, how may I help you?*". Depending on the user's response, the system will then prompt the user with requests for confirmation or further information. Its goal is to carry out a dialog with the user to collect enough information so that the call can be routed to an appropriate destination, such as another automated system or a human operator. The following is an example of such a dialog taken from [22].

Machine : *This is AT&T, how may I help you?*
User : *Can you tell me how much it is to Tokyo?*
Machine : *You want to know the cost of a call?*
User : *Yes, that's right.*
Machine : *Please hold on for rate information.*

In the HMIHY system, an input utterance spoken by a user is first decoded by a continuous speech recognizer. The decoded word string is then passed to the spoken language understanding unit for analysis. This service is designed for any user to call from anywhere and work in real time. It is considered to be a very difficult CSR task. The word error rate for such a system is usually very high. Recognition results will be presented later in this section. A more detailed description of the HMIHY task can be found in [22].

4.1.2 The HMIHY Speech Corpus

The HMIHY corpus was collected over the telephone network from both human-human and human-machine dialog scenarios. The first customer utterances, responding to the greeting prompt of "*This is AT&T, how may I help you?*", were end-pointed, transcribed, and stored. They were partitioned into two sets. A set of 1000 utterances was selected and designated for testing. This set of data will be referred to as the *test1K* data set. These utterances are on average 5.3 seconds in duration and 18 words in length. Another set of 2243 utterances was designated for training the acoustic HMM models used for recognition. This same set of training data will also be used for training the UV models, which include the target, background, and impostor acoustic models. This set of data will be referred to as the *train2K* data set.

In order to provide additional data for training the UV models, six additional data sets were used. These additional data sets are referred to as *greeting, billing method, confirmation, re-prompt, phone number,* and *card number.* They correspond to utterances spoken in response to different prompts in the dialog. They were collected during a more recent evaluation of the HMIHY system relative to the collection of the *test1K* and *train2K* data sets. The kind of utterances each set contains is sufficiently described by the title of each data set, e.g., *billing method, confirmation.* Table 4-1 summarizes some of the statistics of each of the data sets mentioned above.

The characteristics of the six sets of additional training data are quite different from the *train2K* data set for several reasons. First of all, the speech utterances in these data sets are not well end-pointed. Some utterances may contain many seconds of silence or noise proceeding or following speech. Secondly, a large percentage of utterances in some

of the data sets are single word utterances as is evident from the statistics shown in Table 4-1. Finally, since these utterances correspond to different stages in the dialog, there may be a slightly greater training-testing mismatch. Despite these disadvantages, these six sets of data were still used in this experiment for training the target and impostor models.

title of data set	number of utterances	words per utterance	single word utterances	
			frequency	examples
test1K	1000	17.9		
train2K	2243	17.8		
greeting	1769	6.7	632	operator, hello
billing method	1390	3.3	381	collect, card
confirmation	1786	2.0	1502	yes, no
re-prompt	842	9.0		
phone number	793	10.4		
card number	469	9.1		
All training data	9292	8.6		

Table 4-1 Statistics of various data sets.

4.1.3 Recognition Models and Performance

The language model used in recognition has a perplexity of about 16. The size of the lexicon is approximately 3600, which contains all words that appeared in the *train2K* data set. With this lexicon, 30% of the test utterances still contain OOV words due to the highly unconstrained nature of the task [22]. The acoustic models used consist of 52 subword HMM models: one single state model for silence, 40 context independent three-state phone models, and 11 digit models with either 8 or 10 states. A list of the subwords used can be found in Table 4-3 in Section 4.3. The names for the phoneme subwords are taken from the ARPABET [25]. The observation density for each state consists a mixture

of eight diagonal Gaussian densities. The feature vectors are 39 dimensional, including 12 cepstral coefficients, energy, and their 1st and 2nd order differences.

The recognition performance for *test1K*, *train2K* data sets, and all training data are tabulated in Table 4-2. The total error rate is defined to be the sum of substitutions, insertions, and deletions, divided by the total number of word occurrences. There are several features worth pointing out. The six sets of additional training data yielded high insertion rates, as is evident from the high insertion rate for all training data. This is due to the lack of end-pointing and the fact that many utterances contain very few words. The word error rate for test data is 56%, which is very high compared to other more constrained tasks. This recognition performance is not the best that has been achieved on the testing utterances. Using context dependent acoustic models and other techniques, a lower word error rate of about 45% can be achieved. For the *train2K* data set, the error rate is only 46% because it was used in the training of the recognition models.

Data set	correct	substitution	insertion	deletion	total error
test1K	52%	36%	8%	13%	56%
train2K	61%	29%	6%	11%	46%
All training data	61%	31%	23%	8%	62%

Table 4-2 Recognition performance for *test1K*, *train2K* data sets, and all training data.

4.2 Testing Procedure

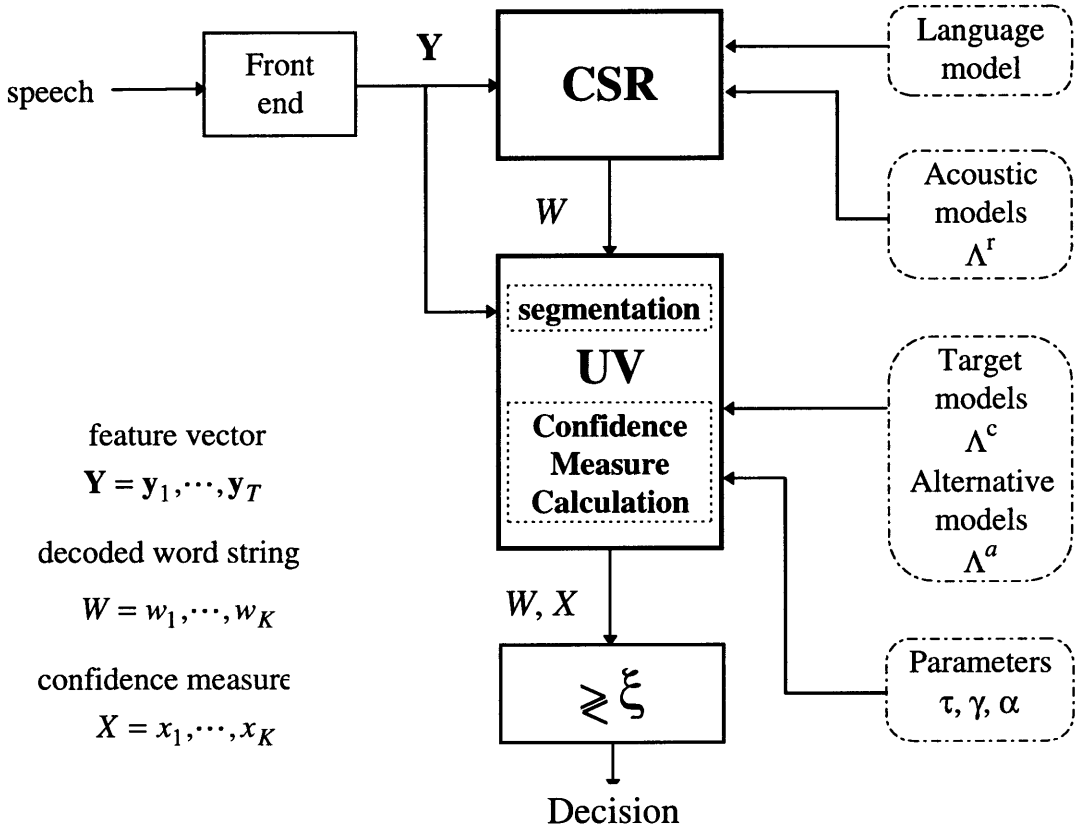

Figure 4-1: Testing procedure for two-pass likelihood ratio based utterance verification.

The testing procedure consists of two main stages, a CSR stage and a UV stage as shown in Figure 4-1. It is performed in several steps:

1. A feature vector sequence, $\mathbf{Y} = \mathbf{y}_1, \cdots, \mathbf{y}_T$, obtained from front end speech processing, is first passed through a CSR system yielding a decoded word string, $W = w_1, \cdots w_K$, using a language model and a set of subword acoustic models, Λ^r.

2. The decoded word string, W, as well as the feature vector sequence, $\mathbf{Y}$, are then passed to the UV unit.

a) Segmentation is performed by a forced alignment of the observation sequence, **Y**, with respect to the target models associated with the word sequence, *W*, to find the optimal state sequence, $q_1,\cdots,q_T$, as defined in Equation (2.4). The state sequence provides a mapping of observation vectors to state indices for the computation of the LR scores given in Equations (3.7)-(3.9). It was found that doing segmentation using the target models, Λ^c, yielded slightly better UV performance than using the recognition models, Λ^r. Note that Λ^r and Λ^c are not identical, even though they are both acoustic models for the same set of subword units. They are different because they are obtained from different training processes.

b) A confidence measure, x_k, is then calculated for each word, w_k, according to Equations (3.7) through (3.11), using the target and alternative models, Λ^c and Λ^a and a set of parameters, τ, γ, and α. Recall that τ and γ are parameters for the sigmoid weighting of the unit level LR scores given in Equation (3.10), and α is the parameter for interpolating the background and impostor hypothesis likelihoods in Equation (3.3). Note that the language model is not used in confidence measure calculation. The confidence measures are purely based on acoustics.

3. Each word level confidence measure is then compared to a threshold, ξ. Decoded words with confidence measure above ξ are accepted and ones with confidence measure below ξ are rejected. The goal is to be able to accept correctly decoded words and reject incorrectly decoded words.

4.3 ML Training Procedure

In the baseline experiments, maximum likelihood training is used for training the UV models, which include a set of target models, Λ^c, and alternative models which consists of the background model, λ^{bg}, and a set of impostor models, Λ^{im}. The background model is a single state 64 mixture HMM model. Its parameters are estimated using an unsupervised ML training procedure using all frames in the *train2K* data set.

The target and impostor models have the same topology as the recognition models, which was described in Section 4.1.3. The two sets of models are trained simultaneously from all training data. The procedure can be outlined as follows:

1. Perform recognition on all utterances in the training data sets.
2. Initialize both target models and impostor models using their corresponding recognition models. For example, the recognition model for subword unit *aa*, λ^r_{aa}, is copied to its corresponding target and impostor models, λ^c_{aa}, and λ^{im}_{aa}.
3. For each utterance, align the decoded transcription with the correct transcription (recognized by human), and label each subword in the decoded word sequence as either correct, insertion, or substitution.
4. Using ML training procedure, train target models and impostor models from units labeled as correct and units labeled as substitution, respectively.

Units labeled as insertions were not used in the training of impostor models due to practical reasons. It is because poor end-pointing and a large percentage of short utterances in the additional training data sets resulted in insertions rates that were too

high. Many of the insertions correspond to silence segments and are not fair representations of acoustic events that are easily confused with particular subword units.

subword	correct	substitution	subword	correct	substitution	subword	correct	substitution
aa	3417	1042	iy	3193	1660	v	423	410
ae	2240	1354	JH	518	260	w	861	424
ah	2117	977	K	11062	2020	y	2857	535
ao	4148	1000	l	8109	2125	z	1474	2224
aw	185	238	m	5146	1628	zh	2	0
ax	6267	1756	n	7098	2107	one	2302	523
ay	6271	1931	ng	2192	869	two	2043	769
B	1802	557	ow	1575	661	three	1899	504
CH	458	232	oy	16	2	four	2266	433
D	6350	2508	P	2407	943	five	1718	315
dh	1157	997	r	5894	1316	six	1776	454
eh	4826	1160	s	4902	1309	seven	1832	398
er	2962	765	sh	306	53	eight	1728	822
ey	2640	1037	T	8989	2543	nine	1382	312
f	1171	425	th	230	378	zero	588	83
G	511	439	uh	193	177	oh	718	502
hh	917	519	uw	2961	1368			
ih	5639	2167	uw	2961	1368	**Total**	**141,738**	**47,231**

Table 4-3 Number of units labeled as correct or substitution for each subword unit in training data. They are directly related to the amount of data used toward training the target and impostor models of each subword unit. The names of the phoneme subwords are taken from the ARPABET [25].

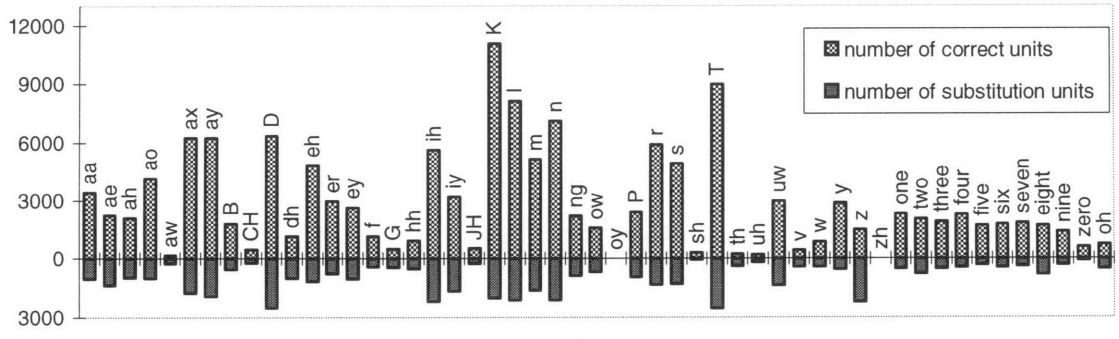

Figure 4-2 Number of units labeled as correct or substitution for each subword unit.

Table 4-3 and Figure 4-2 illustrate the number of units that were labeled as correct or substitution for each subword unit, which is directly related to the amount of data that was used toward training of the target or impostor model of that subword unit. Notice that

there are a lot more units labeled as correct than substitution, which implies that there are less data for impostor model training, which is why we had to use the six additional training data sets. Also, the occurrence count varies greatly across different subword units. Some subwords barely occurred, e.g., oy, zh. This suggests that maybe we could vary the number of parameters to be estimated, that is, using different number of mixtures, for different subword models, which could be considered in future studies.

4.4 Description of Experiments

The baseline experiments investigate three different methods for confidence measure calculation. The purpose of the experiments is to see how UV performance improves as more sophisticated alternative hypothesis models are applied to confidence measure calculation. Each experiment is carried out according to the testing procedure outlined in Section 4.2 unless otherwise noted. The experiments are described in this section, and their results are presented and discussed in Section 4.5. The three experiments involve the use of confidence measures based on *simple log-likelihood scoring*, *likelihood ratio scoring with background model*, and *likelihood ratio scoring with maximum likelihood trained target and impostor models*.

4.4.1 Simple Log-Likelihood Scoring

The first experiment uses a log-likelihood based, rather than likelihood ratio based, confidence measure calculation. It is performed without any of the UV model training outlined in Section 4.3. In this experiment, the recognition models, Λ^r, were used for both segmentation and confidence measure calculation as given in Figure 4-1. To implement

log-likelihood based confidence measure, the frame level score in Equation (3.7), $R_t(\mathbf{y}_t) = \log b^c_{q_t}(\mathbf{y}_t) - \log b^a_{q_t}(\mathbf{y}_t)$, was replaced by $R_t(\mathbf{y}_t) = \log b^r_{q_t}(\mathbf{y}_t)$. When combining unit level scores into word level scores, simple algebraic averaging was used instead of geometric averaging with weighting, that is, Equations (3.10) and (3.11) were replaced with $R_w(\mathbf{Y}_w) = \frac{1}{N}\sum_{u=1}^{N} R_u(\mathbf{Y}_u)$. The purpose of this experiment is to see how well a simple log-likelihood based procedure performs for the purpose of utterance verification.

4.4.2 LR Scoring With Background Model

The second experiment implements a likelihood ratio based confidence measure using only a single state background model as the alternative model. It is performed without the training of either target or impostor models. In this experiment, the recognition models, Λ^r were used for segmentation and as target models in confidence measure calculation. In Equation (3.7), the weight of the background model was set to one, i.e., $\alpha = 1$, so the frame level LR score becomes $R_t(\mathbf{y}_t) = \log b^c_{q_t}(\mathbf{y}_t) - \log b^{bg}(\mathbf{y}_t)$. In the sigmoid weighting of unit level scores in Equation (3.10), the parameters were set to $\tau = 0.0$ and $\gamma = 0.5$.

4.4.3 LR Scoring With ML Trained Target and Impostor Models

The last experiment is the most closely related to the phase II experiments that are presented in Chapter 5. It implements a likelihood ratio based confidence measure using target, impostor, and background models, which were trained according to the maximum

likelihood training procedure outlined in Section 4.3. In this experiment, segmentation was performed using target models and confidence measure calculation was carried out following Equations (3.7)-(3.11) exactly. The parameters in Equations (3.7) and (3.10) were set to $\alpha = 0.2$, $\tau = 0.0$, $\gamma = 0.5$. These parameters were obtained empirically from trial experiments. The choice of α will be discussed again later in Chapter 5.

4.5 Experimental Results

The goal of utterance verification is to accept correctly decoded words while rejecting incorrectly decoded ones. In the baseline experiments, confidence measures were calculated for each word in the decoded word strings. In this section, we will evaluate these confidence measures in terms of their ability to distinguish the two classes of correctly and incorrectly decoded words. This section will first describe the means for this evaluation, then present and discuss the results of the three experiments conducted.

The UV performance for all three experiments was evaluated on only a subset of the words that appeared in the decoded word strings associated with the test utterances. There are two reasons for this. The first reason is that decoded word strings are interpreted by a SLU unit which relies on only a subset of the words and phrases decoded by the recognizer. Thus, it is more important to be able to correctly detect and assign confidence measures to those words and phrases that are considered to be salient by the SLU unit. To select these words, a large list of information bearing phrases was obtained from the SLU training procedure, and words contained in these phrases were extracted. The second reason for evaluating confidence measures on only a subset of the vocabulary is the existence of "short" function words, e.g., *it, a, I, etc.*. These short words tend to be

acoustically unstable, i.e., their acoustic characteristics vary as they appear in different context. It is often the case that these short words are decoded only due to the strength of the language model. Thus, it is not very meaningful to analyze their acoustic confidence measures. Due to this reason, a small set of "short" words were removed from the list. Finally, 134 words were selected, which account for 29% (4905) of the word occurrences in the decoded word strings associated with the 1000 test utterances. They are listed below in alphabetical order.

about, ahead, alternate, another, answered, answering, anyway, area, assist, assistance, because, bill, billable, billed, billing, blind, brazil, business, busy, california, call, called, calling, calls, card, cents, charge, charged, charges, check, checked, city, code, codes, collect, company, completed, connect, connected, cost, country, couple, credit, customer, days, dial, dialed, dialing, different, digit, direct, directory, disconnect, disconnected, distance, dollars, emergency, getting, hang, hello, help, home, hook, hours, house, hundred, hung, incorrect, information, instead, international, italy, line, lines, london, long, looking, make, making, maybe, miles, minutes, misdialed, money, name, name's, number, number's, numbers, off, office, operator, outside, overseas, paid, party, patient, pay, person, phone, pin, place, placed, please, problem, puerto, rico, put, reached, receive, recording, restricted, reverse, rotary, service, several, signal, something, spanish, speak, state, telephone, telling, tennessee, think, third, through, time, touch, universal, visa, washington, whether, wrong

To evaluate the UV performance, the decoded word strings associated with each test utterance is first compared to the correct transcription of that utterances. Each decoded word is then labeled as either correct or incorrect. Incorrectly decoded words include insertions and substitutions, but not deletions, since they do not appear in the decoded word strings. The UV performance will be presented in terms of confidence measure distributions and receiver operating characteristic (ROC) curves.

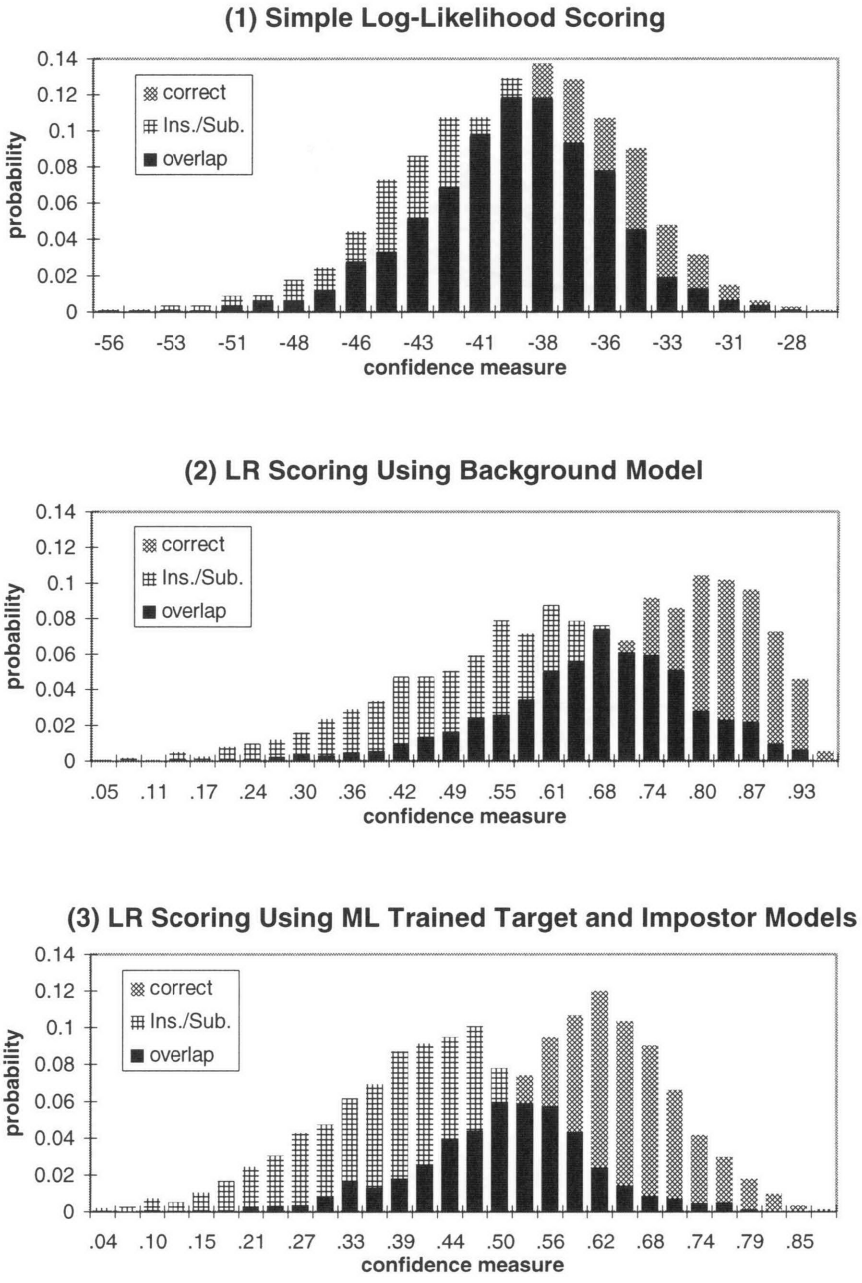

Figure 4-3 Probability distributions of confidence measures for the 2872 correctly decoded words and the 2033 incorrectly decoded words in the test utterances for the three baseline experiments. Overlapped regions are shaded by solid black. Smaller overlapped region indicates better discrimination and better UV performance.

The three pairs of distributions of word level confidence measures for the two classes of correctly and incorrectly decoded words are plotted as histograms and displayed together in Figure 4-3 for comparison. Each histogram consists of 30 equally spaced bins. The x-axis corresponds to the confidence measure, which ranges from 0 to 1 for the LR experiments. The y-axis indicates the probability of a confidence measure being in a particular bin given the word is correctly decoded or incorrectly decoded. The distribution probabilities are obtained by normalizing the word counts within each bin by the total number of words decoded correctly (2872) or incorrectly (2033). The purpose is to eliminate the effect of *a priori* probabilities, $P(C=0)$ and $P(C=1)$, which are determined by the recognition performance. In these plots, the overlapped regions are shaded by solid black. Smaller overlapped region indicates better discrimination and better UV performance. The separation between the means of the two distributions is also a good indicator of performance. The simple log-likelihood scoring experiment had the worst performance. The two distributions almost entirely overlap. In the second experiment when likelihood ratio was employed with a background model serving as the alternative model, the separation improved significantly. The peaks of the two distributions became distinguishable. In the third experiment, where the target models were trained and both the subword dependent impostor models and the subword independent background model were used as the alternative models, the overlap between the distributions continued to decrease.

In Figure 4-4, receiver operating characteristic (ROC) curves representing UV performance are plotted. These curves are plots of probability of detection versus probability of false alarm, generated by sweeping the threshold, ξ, for accepting or

rejecting decoded words. Probability of detection corresponds to the probability of accepting a word given it is correctly decoded. Probability of false alarm is the probability of accepting a word given it is incorrectly decoded. Comparison of the three curves indicates that the UV performance improved significantly.

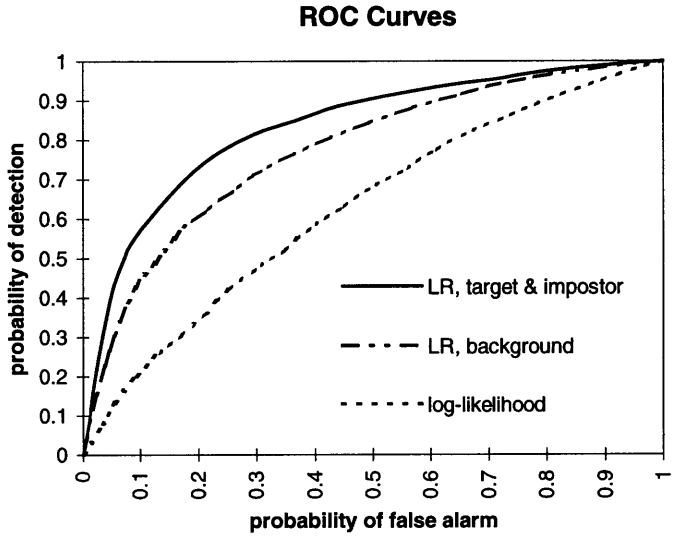

Figure 4-4 Comparison of ROC curves for baseline experiments.

In order to summarize the performance given by a ROC curve with a single number, the measure of equal error rate (EER) is used. It corresponds to the point on the ROC curves where

$$\begin{aligned} & 1 - \textit{probability of detection} \\ = \; & \textit{probability of false rejection} \\ = \; & \textit{probability of false alarm}. \end{aligned}$$

The values of EER for the three experiments are tabulated in Table 4-4. EER decreased by a relative 28% when likelihood ratio with background model was used, and decreased again by a relative 20% when ML training of target and impostor models was employed.

Another measure for UV performance is *efficiency*, which measures the percentage reduction in the entropy of the classification of words, namely,

$$\eta = \frac{H(C) - H(C|X)}{H(C)} \times 100\%, \tag{4.1}$$

in which, $X \in (0,1)$ denotes the confidence measure and $C \in \{0,1\}$ denotes whether the word is correctly decoded or incorrectly decoded. For example, a perfect classifier with $\eta = 100\%$ would have $H(C|X) = 0$. For this case, given the confidence measure, we know deterministically whether the word is correctly decoded or not. On the test data, out of the 4905 decoded words, 2872 (58.6%) were decoded correctly and 2033 (41.4%) were decoded incorrectly, yielding H(C)=0.979. H(C|X) was estimated by quantizing the values of X using 10 uniform bins, namely,

$$H(C|X) = \sum_{i=1}^{10} P\left(X \in \left(\frac{i-1}{10}, \frac{i}{10}\right)\right) \cdot H\left(C \middle| X \in \left(\frac{i-1}{10}, \frac{i}{10}\right)\right). \tag{4.2}$$

The efficiencies for the three baseline experiments are tabulated in Table 4-4. Efficiency increased by an absolute 14% when likelihood ratio with background model was used, and increased again by an absolute 10% when ML training of target and impostor models was employed.

Experiment	EER	Efficiency
Simple log-likelihood scoring	0.408	3.85%
LR scoring with background model	0.294	17.6%
LR with ML trained target and impostor models	0.234	27.2%

Table 4-4 Equal error rates and efficiencies for the three baseline experiments.

4.6 Summary

This chapter described the speech corpus, the experimental procedures for testing and training, the three baseline experiments and their results. The first experiment showed that simple log-likelihood based confidence measures performs poorly in terms of utterance verification. The distributions of confidence measures for correctly decoded and incorrectly decoded words overlaps almost entirely. The second experiment showed that employing a likelihood ratio criterion, even with a very simple definition of alternative hypothesis model, could yield significant improvement in UV performance. The last experiment showed that further improvement in UV performance could be achieved by training the target model and using both the subword dependent impostor models and subword independent background models as the alternative hypothesis models.

5. Phase II: Discriminative Model Training for Utterance Verification

The purpose of this chapter is to evaluate the effectiveness of using a discriminative training procedure for training the utterance verification models. The algorithm of the discriminative training procedure was presented in Section 3.2. It is designed to optimize an LR criterion that is very similar to that used in verification. Therefore, discriminatively trained UV models are expected to out-perform models trained using the ML criterion. The task of this chapter is to verify this expectation and investigate various other aspects of discriminative training. This chapter consists of four major sections. First, the discriminative training procedure used in this experiment is described in detail in Section 5.1. Section 5.2 presents the utterance verification results obtained using discriminatively trained UV models, and compares them to the previous results obtained using ML trained UV models. Section 5.3 discusses the convergence property of the training procedure by presenting the change in UV performance over the training iterations. Section 5.4 investigates various independent issues including the choices of the offset parameter τ and the scaling parameter γ in the definition of the cost function in training. Section 5.4 also examines the sensitivity of the UV performance to the choices of background model weighting parameter α and the word level confidence measure threshold ξ used for accepting or rejecting decoded word hypotheses.

5.1 Discriminative Training Procedure

This section describes steps involved in the implementation of the discriminative training procedure. This training procedure is similar to the ML training procedure described in Section 4.3, except in model parameter re-estimation algorithm. The discriminative training algorithm and detailed formulations were presented in Section 3.2. The training procedure is as follows.

First, speech recognition is performed on utterances in the training data sets described in Section 4.1.1. Each subword unit in the hypothesized word strings is then labeled as being correctly decoded, insertions, or substitutions. These labels are later used in cost computation to assign values to the indicator function, $\delta(u)$, in Equation (3.14). Second, UV models are initialized using the ML trained models obtained in the baseline experiments described in Chapter 4. Third, state segmentation is performed to align the observation frames to the states of the target HMM models. This provides a state assignment so that for each vector $\mathbf{y}_t$, there is $q_t = s_j$, for some state s_j. Finally, each iteration of the iterative training algorithm is performed by estimating the expected cost over the training data and then re-estimate the model parameters. A detailed description of the sequence of the steps taken within the n^{th} iteration is listed below.

I. Estimating Expected Cost:

For each unit labeled as correct or substitution

1. Compute the unit level LR score $R_u(\mathbf{Y}_u)$ [Equation (3.9)].
2. Compute the cost $F_u(\mathbf{Y}_u, \lambda_u^c, \lambda_u^{im})$ [Equation (3.14)].
3. Compute and accumulate gradient of the cost with respect to each model parameter ϕ [Section 3.2.3].

II. Model Parameter Re-estimation:

1. Update the model parameters $\Lambda_{n+1}^{k} = \Lambda_{n}^{k} - \varepsilon \cdot \nabla \overline{F}$ [Equation (3.16)] using different learning rate constants, $\varepsilon_{\mu}, \varepsilon_{\sigma}, \varepsilon_{c}$, for means, variances, and mixture weights, respectively.

2. Update the learning rate constants $\varepsilon_{n} = \varepsilon_{o} \cdot e^{-\rho n}$ where ρ is a positive constant. This exponential decay is chosen to reduce the learning rate as more iterations are taken.

5.2 Experimental Results

This section describes the experimental results obtained for performing UV on the task described in Section 4.1.1 using discriminatively trained UV models. These results are compared to the results obtained using ML trained models in Chapter 4. The UV experiment using discriminatively trained models was performed under the same scenario as described in Figure 4-1. The UV performance will be presented in terms of confidence measure distributions and ROC curves as was done in Section 4.5.

Figure 5-1 is a plot of probability distributions of confidence measures as computed on the test data for correctly decoded and incorrectly decoded words computed using ML trained UV models and discriminatively trained UV models. Compare the two pairs of distributions for discriminatively trained UV models and ML trained UV models, the area in the overlapped region decreased by a relative 6.5% when discriminative training was performed. The plot also shows that the probability distribution of the confidence measures for incorrectly decoded words has shifted leftward. Its mean decreased from 0.42 in the ML experiment to 0.39 in the discriminative training experiment. The mean for correctly decoded words did not change much. This differences in means indicate that the discriminatively trained UV models yielded better separation.

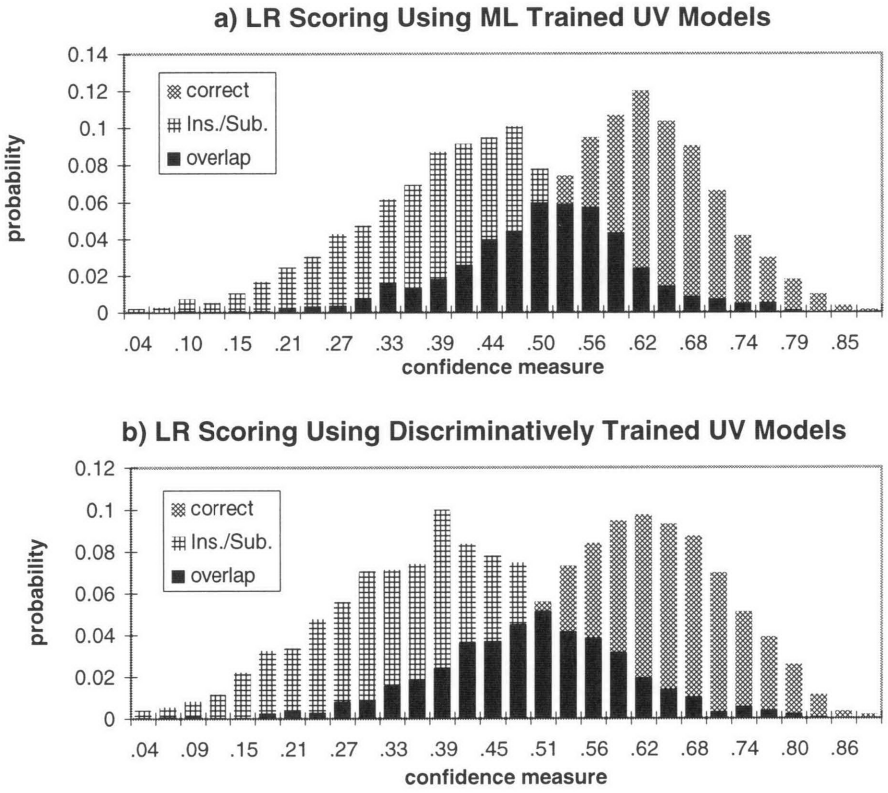

Figure 5-1 Probability distributions of confidence measures for the 2872 correctly decoded words and the 2033 incorrectly decoded words in the test utterances computed using a) ML trained UV models and b) discriminatively trained UV models. Overlapped regions are shaded by solid black. Smaller overlapped region indicates better discrimination and better UV performance.

In Figure 5-2 the ROC curves representing UV performance for discriminatively trained UV models and ML trained UV models are plotted. The curve corresponding to discriminative training is slightly higher, indicating slightly better performance. Table 5-1 tabulates the equal error rates and efficiencies for the two experiments. Comparing the performance of discriminatively trained UV models to ML trained UV model, there is a relative 7.3% decrease in EER and a relative 11% increase in efficiency. In summary, there is some improvement obtained by employing discriminative training in UV. Issues that

may affect the performance of the discriminative training procedure will be discussed in Section 5.3 and 5.4.

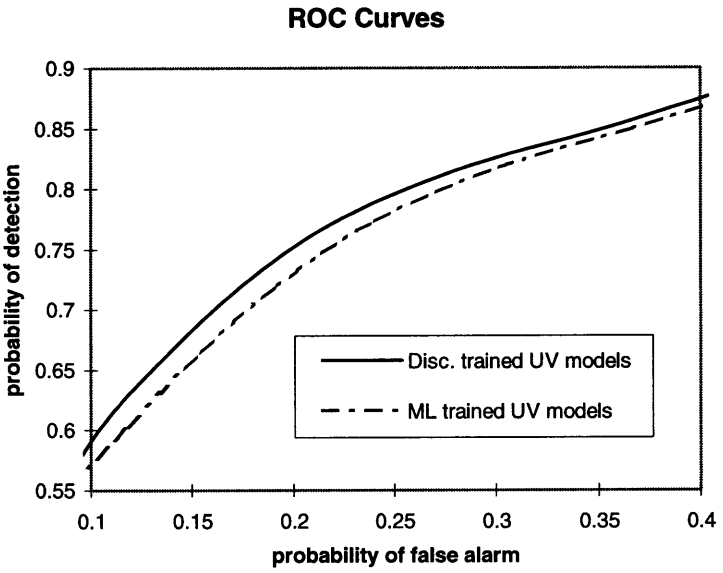

Figure 5-2 ROC curves representing UV performance for discriminatively trained UV models and ML trained UV models.

Experiment	EER	Efficiency
Using ML trained UV models	0.234	27.2%
Using Disc. trained UV models	0.217	30.3%

Table 5-1 Equal error rates and efficiencies for utterance verification using ML trained UV models and discriminatively trained UV models.

5.3 Convergence of Model Training Procedure

The discriminative training procedure is an iterative procedure whose goal is to minimize the average cost function defined in Equation (3.15) for each subword unit. A gradient descent algorithm is used, which is guaranteed to converge to a local minimum of the cost function in the parameter space. This chapter investigates the rate of convergence

of this algorithm on training data and its ability to generalize to unseen test data. Understanding the manner in which the convergence occurs would allow us to better evaluate the discriminative training procedure. This section consists of two parts. The first part focuses on the rate of convergence of the average cost measured on the training data for a single subword unit, *aa*. The second part focuses on how the word level UV performance on training and testing data changes as more iterations of training are used.

Figure 5-3 is a plot of the value of the average cost measured over the training data for subword unit *aa*, i.e., $\overline{F}_{aa}$, as a function of the number of training iterations. Recall that the goal of discriminative training based on the gradient descent algorithm is to minimize this cost. The decrease seen in the plot demonstrates that the algorithm does indeed accomplish this goal. After 25 iterations of training, the value of the average cost has decreased by as much as 50%.

Figure 5-4 displays the means of unit level LR scores for correctly decoded and incorrectly decoded *aa* units after each iteration of the discriminative training procedure. The means are simplified representations of the empirical probability distributions of unit level LR scores for correctly decoded and incorrectly decoded *aa* units, similar to those displayed in Figure 5-1. It is clear from Figure 5-4 that the means of unit level LR scores of correctly decoded *aa* units increased and that of incorrectly decoded units decreased. Together they imply that the distributions of unit level LR scores for correctly decoded and incorrectly decoded *aa* units have drifted apart, which is the desired result.

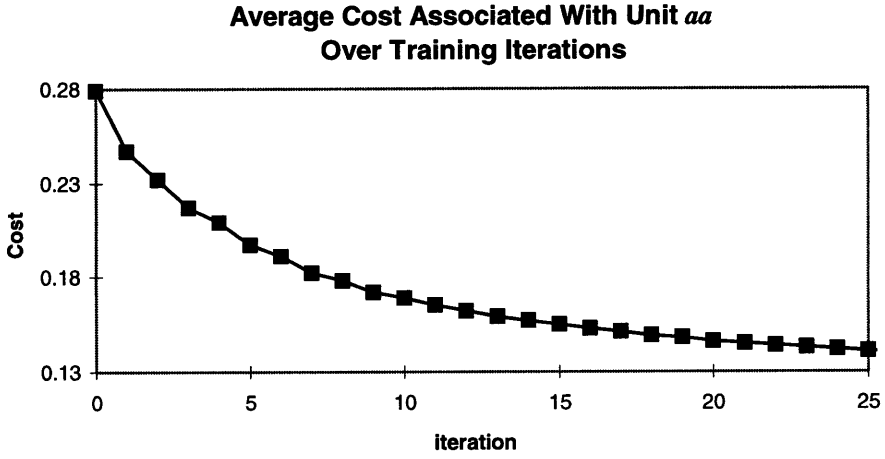

Figure 5-3 Value of average cost measured over the training data for subword unit *aa*, i.e., $\overline{F}_{aa}$, as a function of the number of training iterations.

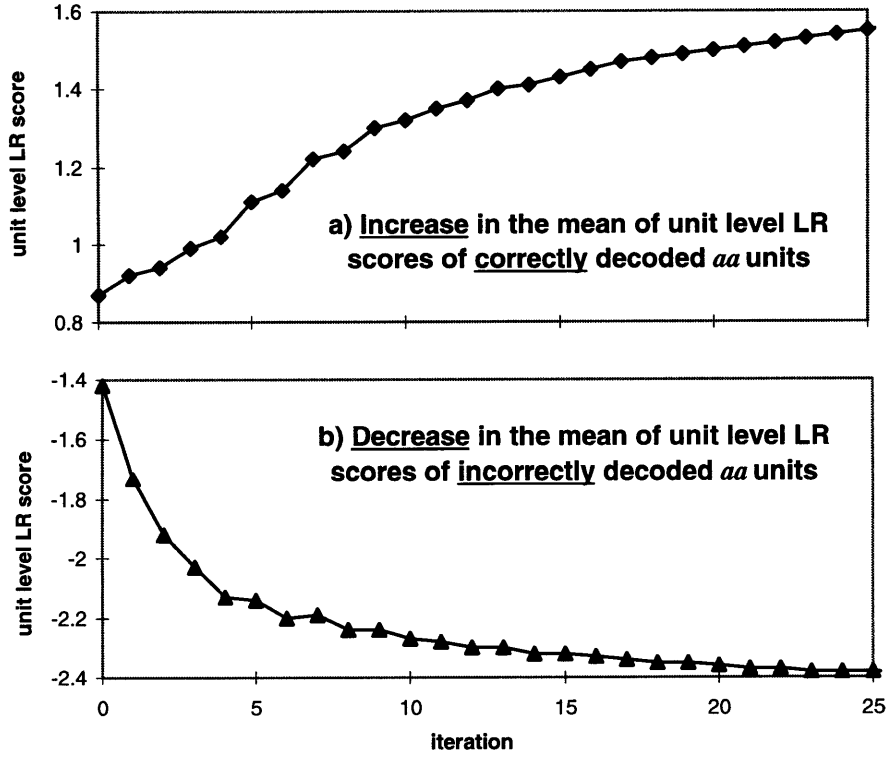

Figure 5-4 Evolution of the means of the empirical distributions of unit level LR scores for correctly decoded and incorrectly decoded *aa* units. Each is a function of the number of training iterations. Together, they demonstrate an increase in separation in unit level LR scores for correctly decoded and incorrectly decoded *aa* units.

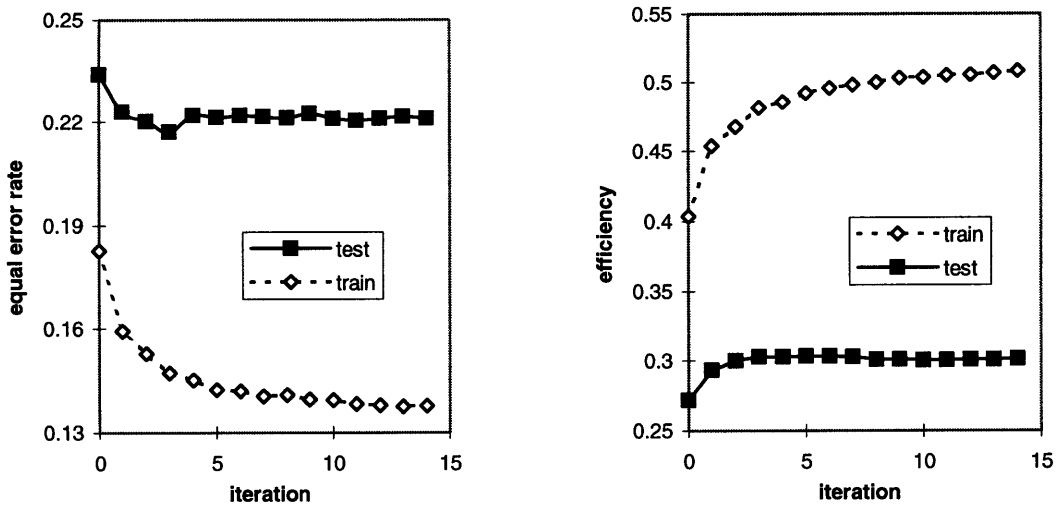

Figure 5-5 Equal error rate and efficiency for utterance verification measured on both testing and training data as a function of the number of training iterations. UV performance on training data continues to improve while performance on testing data stopped improving after three iterations.

Figure 5-5 shows the equal error rates and efficiencies for utterance verification on testing and training data as functions of the number of training iterations. The plots show that the UV performance on training data continues to improve as more iterations of training are used. It is a direct consequence of the increase in separation of the unit level LR scores demonstrated in Figure 5-4. However, UV performance on testing data stopped improving after three iterations. This improvement on training data but not on testing data indicates a possibility of over-fitting. Over-fitting in this context implies that there may be insufficient training examples with respect to the total number of HMM parameters that are being estimated. This is evident from Table 4-3 and Figure 4-2, which presented the amount of data used toward training of target and impostor models of each subword unit. There is also a possibility of mismatch between training and testing data which may be attributed to the extremely wide variety of spontaneous speech utterances that were presented to the system. Recall the description of the speech corpus in Section 4.1.2, some

of the training utterances correspond to different stages in the HMIHY dialog than the testing utterances, which could also contribute to the training-testing mismatch.

5.4 Additional Issues In LR Based Training and Testing

This section discusses three independent issues related to either the training and testing procedure. The first issue is the choice of the cost function parameters, τ and γ, used in discriminative training. The second and third issues are related to the choices of the background model weighting parameter α and the word level confidence measure threshold ξ used in the utterance verification procedure.

Recall the discriminative training procedure described in Section 5.1. In the computation of the cost function F_u as defined in Equation (3.14), the offset parameter was set at $\tau = 0.0$ and the scaling parameter was set at $\gamma = 2.5$. These parameters were determined empirically so that the cost function defined would be effective for the purpose of discriminative training. Recall the plot of the cost function given in Figure 3-2, it is clear that the gradient obtains its maximum value at the LR score that equals to the offset value τ. It was concluded that LR scores close to τ will contribute the most to the gradient computation and consequently have the potential of being affected the most by the training. Therefore, a cost function should be defined so that the offset parameter τ is set roughly between the probability distributions of correctly decoded and incorrectly decoded units, where discrimination is most needed. The scaling parameter γ should reflect the separation between the distributions.

Figure 5-6 is a plot of the means of the unit level LR scores for correctly decoded and incorrectly decoded units and their averages for each subword unit before discriminative training is performed. The means are simplified representations of the probability distributions as were in Figure 5-4. Figure 5-6 shows that there is a large variation in the mean values and their separation across different subword units, which implies a large variation in the probability distributions. Therefore, there would be different optimal choices of τ and γ for each subword unit. For example, a set of near optimal choices of τ is marked by horizontal bars in Figure 5-6. They are averages of the mean unit level LR scores for correctly decoded and incorrectly decoded units. The dotted line in the figure corresponds to the unit independent choice of $\tau = 0.0$ used in this experiment. It can be concluded from the figure that using $\tau = 0.0$ for all units is a reasonable choice. Therefore, in this experiment, unit independent choices of τ and γ were used for simplicity. Unit dependent choices could be investigated in the future.

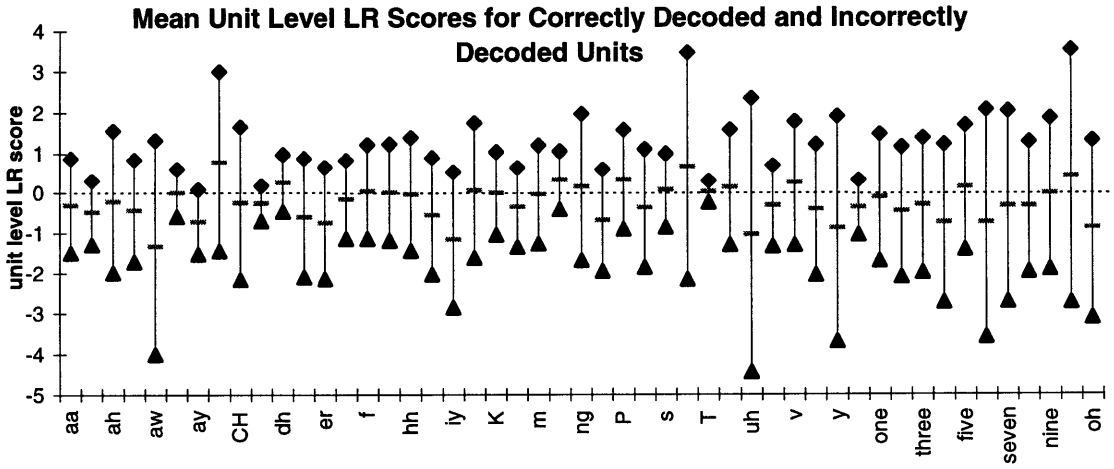

Figure 5-6 Mean unit level LR scores for correctly decoded units (diamond) and incorrectly decoded units (triangle) for each subword unit before starting discriminative training. Their averages are marked by short horizontal bars. The dotted horizontal line across the figure corresponds to the unit independent choice of $\tau = 0.0$.

The next two issues that will be discussed are related to the choices of the background model weighting parameter α and the word level confidence measure threshold ξ used in the UV procedure described in Section 4.2. In the testing experiments performed in this study, the correct transcriptions of the testing utterances are known. Thus, appropriate values of α and ξ can be chosen to optimize the UV performance. However, in field trials, where correct transcriptions are not available, values of α and ξ have to be chosen in advance. The following discussion is dedicated to determining the sensitivity of UV performance to the choices of α and ξ.

Figure 5-7 is a plot of UV performance measures, EER and efficiency, as functions of the background model weighting parameter α defined in Equation (3.3). It shows that for values of α in the neighborhood of 0.2, the performance measures do not vary much. $\alpha = 0.2$ is used in all the experiments performed in this study that involves using impostor models, unless otherwise stated.

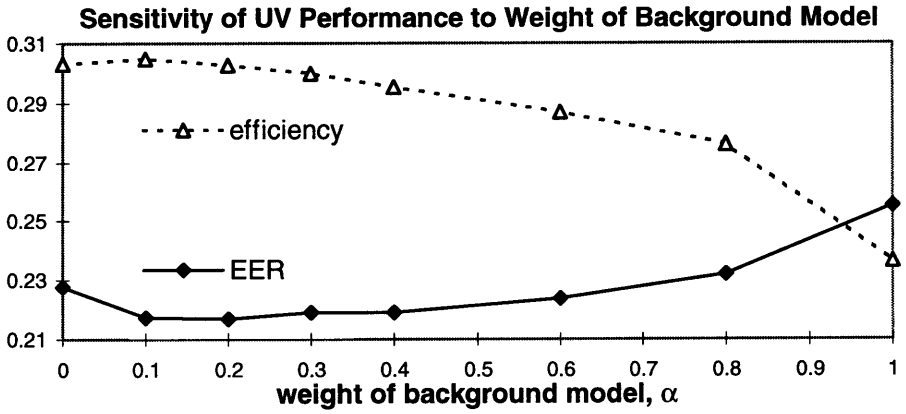

Figure 5-7 Sensitivity of UV performance, in terms of efficiency and EER, to the weight of background model α.

Figure 5-8 is a plot of various verification errors, including probability of false rejection, probability of false alarm, and their average, as functions of the threshold setting ξ as defined in Equation (3.12). As the threshold varies from 0.4 to 0.6, probability of false rejection and probability of false alarm both change significantly, one increases while the other decreases. However, their average error does not vary as much. Comparing to the middle point where all three curves intersect, which corresponds to equal error rate, the average error increases by about a relative 10% as the threshold deviates by 0.05, and increases by about a relative 30% when the threshold is off by 0.1. In most cases, setting the threshold around 0.5 should be near optimal.

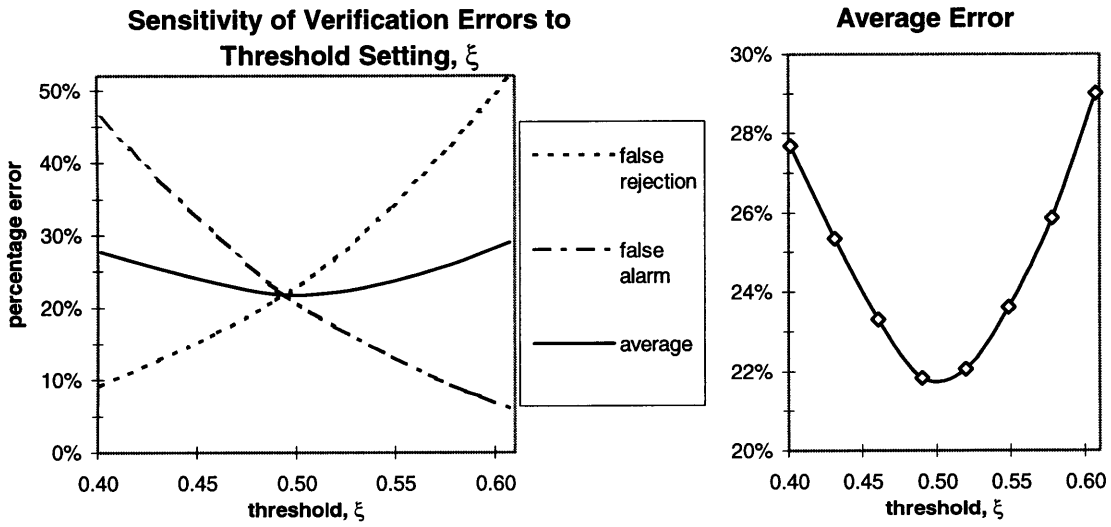

Figure 5-8 Sensitivity of verification errors to threshold setting, ξ. The plot on the right is a close up of the middle curve in the plot on the left.

5.5 Summary

The purpose of this chapter is to evaluate the effectiveness of using a discriminative training procedure for utterance verification model training. Section 5.1 described the training procedure and Section 5.2 presented the results and compared them to the results obtained using ML trained UV models. It was shown that employing discriminative training procedure resulted in a relative 7.3% reduction in equal error rate, which corresponds to an improvement in UV performance. Section 5.3 discussed the convergence of the training process. It is shown that although UV performance on training data significantly improved over iterations, UV performance on testing data stopped improving after three iterations of training. This phenomenon suggests a possibility of over-fitting, which in this context implies that there may be insufficient training data. Section 5.4 discussed three other issues. It is shown that different subword units have different probability distributions of unit level LR scores for correctly decoded and incorrectly decoded units. Therefore, there might be an advantage to using unit dependent offset parameter τ and scaling parameter γ for cost function definition in training. It is also shown that the UV performance is not very sensitive to the choices of background model weighting parameter α and is also not very sensitive to word level confidence measure threshold ξ when the deviation from optimal value is small.

6. Phase III: Further Applications of Utterance Verification

This chapter presents three experimental studies of issues that are related to further applications of acoustic confidence measures. The first experiment investigates using sentence level confidence measures for rejecting utterances that contain only background noise, silence, or non-speech utterances. The second experiment compares the UV performance for assigning confidence measures at a phrase level versus assigning them at the word level. The last experiment implements a method for converting the word level LR based confidence measure of a decoded word to an estimate of the *a posteriori* probability of the word being correctly decoded given its confidence measure. In all the experiments presented in this chapter the utterance verification models used are trained using the discriminative training procedure described in Chapter 5.

6.1 Sentence Level Utterance Verification

This section describes an experiment which investigates using sentence level confidence measures for the purpose of rejecting utterances that contain only background noise, silence, or non-speech utterances. They will be referred to collectively as garbage utterances. This section is divided into three parts. The first part describes the garbage utterances and motivates the need for effective rejection of these utterances. Second, the experimental setup is described. Finally, the results of a sentence level UV experiment are presented.

6.1.1 Problem Description

In a telephone based spoken language understanding task like the HMIHY task, there are many situations where garbage utterances are interpreted as speech utterances resulting in unpredictable actions being taken. A user may be silent but the telephone receiver may pick up ambient noise, which may be interpreted as a speech utterance. A user may enter a string of digits using the number keys on a touch-tone telephone instead of speaking the digits and the recognizer may attempt to interpret the touch-tone digits as a speech utterance. It is very important to be able to separate garbage utterances from true speech. After a garbage utterance is passed to an ML based recognizer, the recognizer would reject the utterance only when it is unable to produce a hypothesized word string because no allowable network path could be found in the search procedure. The speech recognizer may accept the garbage utterance and produce a hypothesized word string. This meaningless word string will then be passed to the SLU unit and result in unpredictable actions being taken. Therefore, it is necessary to develop a mechanism that can reliably reject these garbage utterances.

6.1.2 Experimental Setup

This experiment is performed using the 1000 utterances in the *test1K* data set and another data set in the HMIHY speech corpus that contains exclusively garbage utterances. A total of 784 garbage utterances were presented to the recognizer. The recognizer was unable to produce hypothesized word strings for only about 30% of the utterances. Hypothesized word strings were produced for the remaining 543 of the garbage utterances. Without any means for verifying these hypothesized strings, they

would be passed directly to the SLU unit. We attempt to reject these garbage utterances that were decoded by the recognizer by assigning them sentence level confidence measures which can then be compared to a threshold. The sentence level confidence measure of an utterance is defined to be the algebraic mean of the word level confidence measures assigned to each word in the decoded word string, i.e.,

$$R_S = \frac{1}{K_S} \sum_{k=1}^{K_S} R_{w_k}(\mathbf{Y}_{w_k}), \tag{6.1}$$

where K_S is the total number of words in a decoded word string , and $R_{w_k}(\mathbf{Y}_{w_k})$ denotes the word level confidence measures defined in Equation (3.11).

6.1.3 Experimental Results

To evaluate the ability of the sentence level confidence measures to correctly identify garbage utterances, sentence level confidence measures were computed for the decoded word strings associated with all 543 garbage utterances and the 1000 utterances in the *test1K* data set. The goal is to correctly reject the garbage utterances while accepting the *test1K* utterances. We evaluate the rejection performance by plotting the probability distributions of the sentence level confidence measures for the *test1K* and garbage utterances and the ROC curve.

Figure 6-1 is a plot of probability distributions of sentence level confidence measures for the 1000 *test1K* utterances and the 543 garbage utterances. The overlapped region is small, which indicates good discrimination. Figure 6-2 is the plot of the ROC curve. Probability of detection corresponds to the probability of accepting an utterance given that it is one of the *test1K* utterances. Probability of false alarm corresponds to the

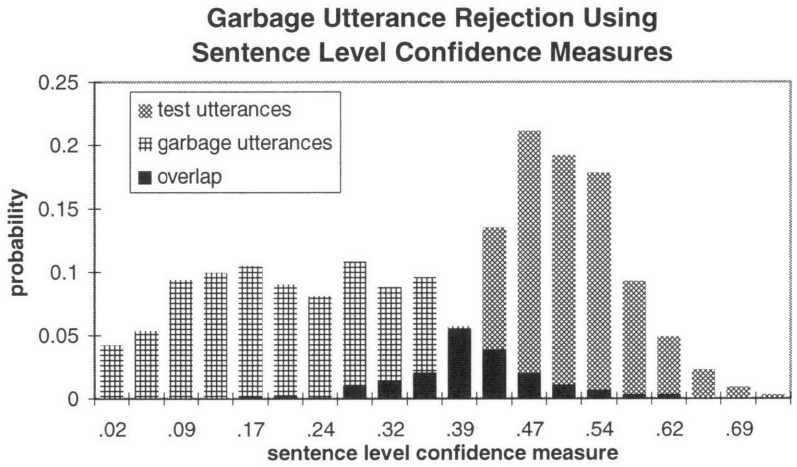

Figure 6-1 Probability distributions of sentence level confidence measures for the 1000 *test1K* utterances and the 543 garbage utterances.

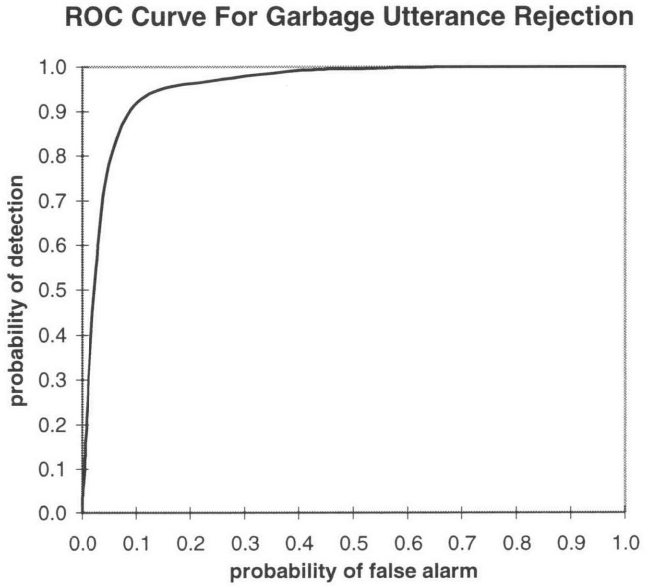

Figure 6-2 ROC curve representing performance of garbage utterance rejection using sentence level confidence measures.

EER	0.097
efficiency	64.7%

Table 6-1 Equal error rate and efficiency for garbage utterance rejection using sentence level confidence measures.

probability of incorrectly accepting an utterance given that it is one of the garbage utterances. The plot shows that low false alarm rates can be achieved at low risk of false rejection. In particular, the equal error rate is 0.097, which is presented in Table 6-1. The efficiency for using sentence level confidence measures for sentence level utterance verification is 64.7%, which is also presented in Table 6-1. This implies that the sentence level confidence measure is able to provide a lot of information for whether an utterance that has already been decoded by the recognizer is a garbage utterance.

6.2 Phrase Level Utterance Verification

This section describes an initial attempt to use confidence measures for utterance verification at a phrase level. There are two reasons for exploring phrase level UV. One reason is that the confidence measures calculated are passed to a SLU system that interprets the semantic meaning of a decoded utterance based on phrase level units, e.g., *make a credit card call.* Therefore, it may be useful to assign confidence measures at a phrase level so that the confidence measures may be better incorporated into the decision making process of the SLU. The other reason is that the acoustic characteristics of a word are often affected by its surrounding context, especially for shorter words. Assigning confidence measures at a phrase level reduces this effect by taking some context into consideration and at the same time increasing the length of the speech segment. This section will first describe how phrase level confidence measures are defined and then evaluate the technique on a single phrase,*credit card.*

The definition of a phrase level confidence measure is very similar to that of a word level confidence measure defined in Equation (3.11). It is the geometric mean of the weighted unit level LR scores of all subword units contained in the phrase,

$$R_{ph}(\mathbf{Y}_{ph}) = \exp\left(\frac{1}{N_{ph}}\sum_{i=1}^{N_{ph}}\log\left(F_{u_i}(\mathbf{Y}_{u_i})\right)\right), \tag{6.2}$$

where $\mathbf{Y}_{ph}$ is the sequence of feature vectors corresponding to the phrase, N_{ph} is the total number of units contained in the phrase, and $F_{u_i}(\mathbf{Y}_{u_i})$ denotes the weighted unit level confidence measure defined in Equation (3.10).

	Number of Occurences in Test Data	
	correctly decoded	incorrectly decoded
credit	70	70
card	69	71
credit card	66	74

Table 6-2 Number of correctly decoded and incorrectly decoded occurrences of the words *credit* and *card* and the phrase *credit card* in test data.

The practical benefits of computing confidence measures at the phrase level are demonstrated here for the phrase *credit card*. There are a total of 140 occurrences of the phrase in the hypothesized word strings associated with the 1000 test utterances. Using confidence measure calculation procedures similar to that described in Section 3.1, word level and phrase level confidence measures were computed for each of the 280 hypothesized words and 140 hypothesized phrases. Table 6-2 demonstrates how many of the words and phrases were decoded correctly and how many were decoded incorrectly. A phrase is considered to be correctly decoded only when all of the words contained in the phrase are correctly decoded.

	EER	efficiency
credit	.143	54.6%
card	.188	41.0%
credit card	.122	59.1%

Table 6-3 EERs and efficiencies for using word level confidence measures on the words *credit* and *card*, and using phrase level confidence measures on the phrase*credit card.*

To evaluate the UV performance for each of the words, *credit*, *card*, and the phrase *credit card*, equal error rates and efficiencies are computed and tabulated in Table 6-3. It is clear from the table that assigning confidence measures at the phrase level yielded better UV performance than assigning confidence measures at the word level. Other similar trials were performed on phrases such as *long distance*, *to call*, etc. Various degrees of improvement were observed. For the phrase *long distance*, the phrase level UV performance was better than both word level UV performance, same as the phrase *credit card*. However, for the phrase *to call*, the UV performance improved for the word *to*, but degraded for the word *call*. This could be due to the highly variable acoustic characteristics of the short word *to*. A systematic way of choosing phrases and applying phrase level confidence was not fully investigated, which could be considered in the future.

6.3 Obtain *A posteriori* Probability From Confidence Measure

This section investigates a method for converting the word level LR based confidence measure for each decoded word to an estimation of the *a posteriori* probability of the word being correctly decoded given its confidence measure. This conversion is motivated by the fact that these acoustic confidence measures are passed to a SLU system that is based on a probabilistic framework. In this section, a method for estimating these *a*

posteriori probabilities based on empirical distributions of confidence measures is described. The manner in which these *a posteriori* probabilities are incorporated into the SLU system is described in [20].

The *a posteriori* probability of a word being correctly decoded given its confidence measure can be expressed as

$$P(C=1 \mid X=x) \quad \text{for } x \in (0,1), \tag{6.3}$$

where $C=1$ corresponds to the event of a word being correctly decoded and $X=x$ corresponds to the event of the confidence measure of the word being x. In this work, as an initial attempt, we will approximate $P(C=1 \mid X=x)$, which is a continuous function of x, with a discrete function by quantizing the values of x. In particular, the range of the confidence measure (0,1) is partitioned into ten equally sized intervals as shown in Figure 6-3. Each interval can be visualized as a *bin*. Then, instead of estimating the probabilities defined in Equation (6.3), we will estimate a set of ten discrete probabilities,

$$P\left(C=1 \mid X \in \left(\frac{i-1}{10}, \frac{\mathrm{i}}{10}\right)\right) \text{ for } i=1,\cdots,10. \tag{6.4}$$

In this work, a set of probabilities defined in Equation (6.4) is estimated for each word in the lexicon due to the empirical observation that these *a posteriori* probabilities vary from word to word.

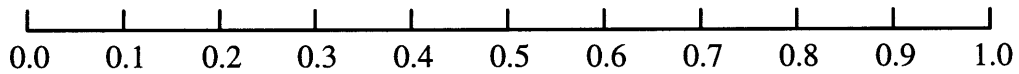

Figure 6-3 Quantization of the range of confidence measures into ten equally sized *bins*.

The *a posteriori* probabilities in Equation (6.4) are estimated empirically from statistics collected over the *train2K* data set. A non-parametric method is used. The idea is to estimate $P\left(C=1 \mid X \in \left(i-1/10,\ i/10\right)\right)$ for a word w simply by going through the training data and counting the fraction of words decoded correctly, among all occurrences of the word w with confidence measure $X \in \left(i-1/10,\ i/10\right)$. An outline of the procedure is as follows.

First, word level confidence measures are computed for each word in the decoded word strings associated with the *train2K* utterances. Next, for each word w in the lexicon, empirical distributions of word level confidence measures for correctly decoded and incorrectly decoded occurrences of the word w are accumulated. The empirical distributions are accumulated as histograms over the same ten bins that are used for quantizing the confidence measure x. As an example, the empirical distributions for the word *my* are plotted in Figure 6-4 (a). It can be read from the plot that among the occurrences of the word *my* with confidence measure within the 5th bin (0.4, 0.5), 97 were decoded correctly and 123 were decoded incorrectly.

The last step in estimating the probabilities in Equation (6.4) is to compute the fraction of words decoded correctly for each bin of each word. For example, for the 5th bin of the word *my*,

$$P(C=1 \mid X \in (0.4,\ 0.5)) \approx \frac{97}{97+123} = 0.44 . \qquad (6.5)$$

In Figure 6-4 (b), the estimated discrete *a posteriori* probabilities for the word *my* is plotted as a function of confidence measure. It is the mapping function for converting

confidence measures to *a posteriori* probabilities for the word *my*. When estimating the *a posteriori* probability for the last bin, due to the low number of occurrences as seen in Figure 6-4 (a), a linear interpolation is used. For words with very low overall number of occurrences, it is not practical to train word dependent *a posteriori* probabilities for each word. Instead, a set of default probabilities is trained from all words.

Theoretically, a set of development data different from both training data used for UV model training and testing data should have been used for the training of the *a posteriori* probabilities. However, additional data was not available. Furthermore, there are various problems associated with quantization and estimation that are not treated in this simple training method. More sophisticated training method could be investigated in future works.

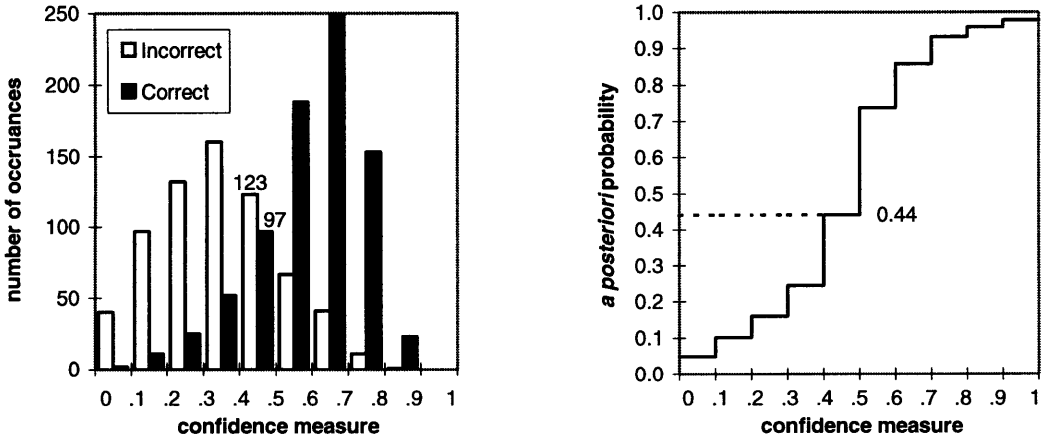

Figure 6-4 Left: (a) Empirical distributions of confidence measures for the word *my*. Right: (b) Estimated discrete *a posteriori* probabilities for the word *my*.

Given the mapping functions from confidence measures to *a posteriori* probabilities for each word in the lexicon, such as the one plotted in Figure 6-4 (b), the word level LR based confidence measures can then be directly converted to *a posteriori* probabilities of each word being correctly decoded. These *a posteriori* probabilities can be compared to a

threshold for deciding whether to accept or reject a decoded word. To evaluate their UV performance, equal error rate and efficiency were computed and presented in Table 6-4 and are compared to the UV performance of LR based word level confidence measures. It is shown that converting LR based confidence measures to *a posteriori* probabilities improves the UV performance slightly. The reason is that different words have different confidence measure distributions, and thus, have different optimal threshold values. Therefore, using one threshold for all words can only achieve a near optimal performance. Converting confidence measures to *a posteriori* probabilities re-centers the distributions in a way that reduces this variation, and consequently improves the performance. Another advantage for using *a posteriori* probabilities is that it is less sensitive to threshold setting than LR based confidence measures.

	EER	efficiency
LR based confidence measures	.217	30.3%
a posteriori probabilities	.211	32.2%

Table 6-4 EERs and efficiencies for utterance verification using LR based confidence measure and using *a posteriori* probabilities converted from confidence measures.

6.4 Summary

This chapter has presented the results of three experimental studies related to further applications of acoustic confidence measures. The first experiment investigated garbage utterance rejection using sentence level confidence measures. In discriminating garbage utterances from legitimate utterances, an EER of 0.097 was achieved. The second experiment investigated the potential of phrase level confidence measures. For the phrase *credit card*, it was shown that assigning phrase level confidence measures yielded better UV performance than assigning word level confidence measures to each individual word

in the phrase. The last experiment investigated a simple method for converting LR based confidence measures to *a posteriori* probabilities of words being correctly decoded. The procedure for estimating non-parametric discrete word dependent mapping functions for this conversion by collecting statistics of confidence measures from training data is described. It is shown that *a posteriori* probabilities yielded slightly better UV performance on test data than LR based confidence measures.

7. Conclusions

7.1 Summary

In this thesis we developed and evaluated a likelihood ratio (LR) based utterance verification (UV) procedure and a discriminative training procedure for training a dedicated set of hidden Markov models (HMM) that are used for UV. The UV procedure verifies the accuracy of each decoded word in hypothesized word strings produced by a HMM based CSR decoder. In the verification process, the UV system assigns each decoded word a word level confidence measure that is formed from a nonlinear combination of subword level LR scores. Each LR score for each subword unit is computed using a target hypothesis model and an alternative hypothesis model dedicated for each subword unit. The alternative hypothesis density is a linear combination of a subword independent *background* density for representing the generic spectral characteristics of speech and a set of subword dependent *impostor* HMM densities for representing subword specific variabilities. The discriminative training procedure is designed based on a gradient descent algorithm with a LR criterion similar to that used in verification. It is an iterative procedure which re-estimates the UV model parameters to increase the separation between the LR scores for correctly decoded and incorrectly decoded subword units.

The LR based UV procedure and the discriminative training procedure are evaluated on 1000 utterances collected from a highly unconstrained large vocabulary spoken

language understanding task performed over the public telephone network. The UV performance was measured in terms of its ability to accept correctly decoded words while rejecting incorrectly decoded ones. Confidence measure distributions for correctly decoded and incorrectly decoded words were plotted and receiver operating characteristic curves were generated to compare the relative merits of different UV implementations. Baseline experiments described in Chapter 4 demonstrated that LR based confidence measures computed using both subword dependent impostor models and a single state subword independent background model as the alternative models yielded significantly better UV performance than using only the subword independent background model as the alternative model. It was also shown that all LR based UV model definitions that were investigated here significantly out-performed attempts at using the unnormalized likelihood score obtained using an ML criterion.

Phase II experiments were related to discriminative training of the UV models, and were described in Chapter 5. It was shown that a relative decrease of 7.3% in equal error rate was obtained, which corresponds to an improvement in UV performance. Study of the convergence rate of this iterative training procedure showed that the UV performance on training data continued to improve while the UV performance on testing data no longer improved after several iterations. While this behavior is typical of many discriminative training procedures, it may also suggest a possibility of over-fitting to the training data.

Chapter 6 presented three additional experiments performed in order to investigate applications beyond word level utterance verification. In the first experiment, sentence level confidence measures were demonstrated to yield good performance when used for rejecting utterances that contain only background noise, silence, or non-speech utterances.

Each sentence level confidence measure is the simple algebraic mean of the word level confidence measures assigned to all the words in a decoded word string. The second experiment investigated the possibility of estimating confidence measures at a phrase level rather than the word level. This is motivated by the belief that longer segments like phrases may be less affected by the surrounding acoustic context than word level segments. When evaluated on an anecdotal phrase *credit card*, phrase level confidence measures were shown to yield better UV performance than word level confidence measures. The last experiment investigated a simple method for converting LR based confidence measures to *a posteriori* probabilities of words being correctly decoded given their confidence measures for the purpose of integrating acoustic confidence measures with the statistical formalism associated with the spoken language understanding system.

7.2 Future Work

This thesis has demonstrated a promising potential for using utterance verification in large vocabulary continuous speech recognition. However, there are still various issues not yet fully investigated. Additional effort in those aspects may lead to further improvement in UV performance. First of all, the study of convergence rate in the discriminative training procedure has suggested a possibility of insufficient training data. As more data becomes available in the future, using additional data to train the UV models may yield improvement in UV performance. Secondly, in this experiment, a particular HMM model topology was chosen for practical reasons. There is a great possibility that there might be some advantage to using some other HMM model topologies. Thirdly, in the definition of the cost function used in discriminative training, subword independent

values of the offset parameter τ and the scaling parameter γ were used. It was suggested in Section 5.4 that there might be an advantage to using subword dependent values, and this may be investigated in future work.

Another possibility for improving the UV performance is to modify the segmentation procedure in utterance verification, a step which takes place before the confidence measure calculation as shown in Figure 4-1. The current segmentation procedure uses the Viterbi algorithm which is based on a maximum likelihood criterion. It searches through all possible state sequence to find the one that yields a maximum likelihood score. It would be a reasonable experiment to try performing segmentation based on a LR criterion similar to the one used in confidence measure calculation using a modified Viterbi algorithm as proposed in [18]. This modified decoder searches for a path that yields a maximum likelihood ratio score as opposed to maximizing a likelihood score as done in a conventional Viterbi decoder.

Another set of future experiments, some of which are already in progress, is to integrate utterance verification with language modeling and spoken language understanding, which was described in Section 3.3. Previous language modeling and language interpretation techniques treat all words as if they were all decoded with equal confidence. In the end, the principle motivation for developing robust measures of acoustic confidence is the hope that they can provide a vehicle for inserting acoustic knowledge into statistical language modeling and dialog control.

References

[1] L. Rabiner and B. H. Juang, "Fundamentals of speech recognition," *Prentice Hall Signal Processing Series*, [1a] pp. 69-97; [1b] pp. 435-439; [1c] pp. 337-342; [1d] pp. 449-450, 1993.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, 39 (1):1-38, 1977.

[3] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 129-132, April 1990.

[4] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. R. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans on Acous. Speech and Sig. Proc.*, vol. 38, no. 11, pp. 1870-1878, 1990.

[5] P. Jeanrenaud, J. R. Rohlicek, K. Ng, and H. Gish, "Phonetic-based word spotter: various configurations and applications to event spotting," *Proc. European Conf. on Speech Communications*, September, 1993.

[6] M. Weintraub, "Keyword spotting using SRI's Decipher large vocabulary speech recognition system," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, April, 1993.

[7] R. C. Rose, "Keyword detection in conversational speech utterances using hidden Markov model based continuous speech recognition," *Computer Speech and Language*, vol. 9, no. 4, pp. 309-333, 1995.

[8] R. C. Rose, "Word spotting from continuous speech utterances," *Automatic Speech and Speaker Recognition - Advanced Topics*," C. H. Lee, F. K. Soong, and K. K. Paliwal, editors, Kluwer, pp. 303-330, 1996.

[9] S. R. Young and W. H. Ward, "Recognition confidence measures for spontaneous spoken dialog," *Proc. European Conf. on Speech Communications*, pp. 1177-1179, September, 1993.

[10] M. Weintraub, "LVCSR log–likelihood ratio scoring for keyword spotting," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 297-300, April, 1995.

[11] S. Cox and R. C. Rose, "Confidence measures for the Switchboard database," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 511-514, May, 1996.

[12] Z. Rivlin, M. Cohen, V. Abrash, and T. Chung, "A phone-dependent confidence measure for utterance rejection," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 515-517, May, 1996.

[13] M. G. Rahim, C. H. Lee, and B. H. Juang, "Robust utterance verification for connected digits recognition," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 285-288, May, 1995.

[14] R. C. Rose, "Discriminant wordspotting techniques for rejecting non-vocabulary utterances in unconstrained speech," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, March, vol. 2, pp. 105-108, 1992.

[15] C. Torre and A. Acero, "Discriminative training of garbage model for non-vocabulary utterance rejection," *Proc. Int. Conf. on Spoken Lang. Processing*, June, 1994.

[16] R. C. Rose, B. H. Juang, and C. H. Lee, "A training procedure for verifying string hypotheses in continuous speech recognition," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 281-284, April, 1995.

[17] M. Rahim, C. Lee, B. Juang, and W. Chou, "Discriminative utterance verification using minimum string verification error training," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 3585-3588, May, 1996.

[18] E. Lleida and R. C. Rose, "Efficient decoding and training procedures for utterance verification in continuous speech recognition," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 507-510, May, 1996.

[19] C. V. Neti, S. Roukos, and E. Eide, "Word-based confidence measures as a guide for stack search in speech recognition," *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 883-886, April, 1997.

[20] R. C. Rose, H. Yao, G. Riccardi, and J. H. Wright, "Integrating multiple knowledge sources for utterance verification in a large vocabulary speech understanding system," *Proc. 1997 IEEE Workshop on Speech Recognition and Understanding*, December 1997.

[21] E. Lleida and R. C. Rose, "Utterance Verification in Continuous Speech Recognition: Decoding and Training Procedures," to be published.

[22] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may I help you," to appear in *Speech Communication*, 1997.

[23] J. K. Chen and F. K. Soong, "An N-best candidates-based discriminative training for speech recognition applications," *IEEE Trans on Speech and Audio*, vol. 2, pp. 206-216, January 1994.

[24] W. Chou, B. H. Juang, and C. H. Lee, "Segmental GPD training of HMM based speech recognizer", *Proc. Int. Conf. on Acoust., Speech, and Sig. Processing*, pp. 473-476, April 1992.

[25] J. E. Shoup, "Phonological Aspects of Speech Recognition," *Trends in Speech Recognition,* W.A. Lea, editor, Prentice-Hall, pp. 125-138, 1980.