Discriminative Utterance Verification for Connected Digits Recognition

Mazin G. Rahim, *Member, IEEE,* Chin-Hui Lee, *Senior Member, IEEE,* and Biing-Hwang Juang, *Fellow, IEEE*

***Abstract*—Utterance verification represents an important technology in the design of user-friendly speech recognition systems. It involves the recognition of keyword strings and the rejection of nonkeyword strings. This paper describes a hidden Markov model-based (HMM-based) utterance verification system using the framework of statistical hypothesis testing. The two major issues on how to design keyword and string scoring criteria are addressed. For keyword verification, different alternative hypotheses are proposed based on the scores of *antikeyword* models and a general acoustic *filler* model. For string verification, different measures are proposed with the objective of detecting nonvocabulary word strings and possibly erroneous strings (so-called putative errors). This paper also motivates the need for discriminative hypothesis testing in verification. One such approach based on minimum classification error training is investigated in details. When the proposed verification technique was integrated into a state-of-the-art connected digit recognition system, the string error rate for valid digit strings was found to decrease by 57% when setting the rejection rate to 5%. Furthermore, the system was able to correctly reject over 99.9% of nonvocabulary word strings.**

I. Introduction

DURING recent years, it has become increasingly essential to equip speech recognition systems with the ability to accommodate spontaneous speech input. Although providing this capability facilitates a friendly user-interface, it also poses a number of new problems, such as the inclusion of out of vocabulary words, false starts, disfluency, and acoustical mismatch. For example, in a recent connected digits trial conducted in Bloomington, MN, users who were prompted to repeat their telephone number would often begin by saying "Uh,...," "My telephone number is...," or "what?" In these situations, a speech recognition system must be able to detect and recognize the "keywords" and reject the "nonkeywords." Recognizers equipped with a *keyword spotting* capability allow users the flexibility to speak naturally without the need to follow a rigid speaking format.

Significant progress has been made in keyword spotting for unconstrained speech using hidden Markov models (HMM's). Keyword spotting systems introduce a filler (or garbage) model for enhancing keyword detection and absorbing out-of-vocabulary events. Proper modeling of filler models using out-of-vocabulary events is essential for improving the performance of a general keyword spotting system. The issue of how to build an appropriate filler model has been extensively studied during the past few years. Rohlicek *et al*. [24] described a keyword spotting system based on a continuous-density HMM with a filler that was constructed from either segments of the keyword models or by weighting the distributions in the keyword states. Wilpon *et al*. [37] presented a keyword spotting system that used a single filler model which was developed by training on transmission noise and extraneous speech input.

To reduce false alarm rate, a large number of studies have incorporated *keyword verification* following detection and segmentation of speech into keyword hypothesis via a conventional Viterbi search. These studies employ some type of a *likelihood ratio* distance to verify whether or not a given keyword exists within a segment of speech. The key issue is the selection of an appropriate alternative model to provide an antiword scoring in computing the likelihood ratio statistics. This has been traditionally done using a general acoustic filler model. Rose and Paul [27] reported high performance keyword verification with monophone filler models trained from transcribed speech. Bourlard *et al.* [3] obtained improved keyword verification performance over a single monophone filler model or multiclass broad phonemic models by using the average of the N-best local scores of the phonemic models as an antiword scoring. Besides using likelihood scores, Moreno *et al.* [16], Chigier [5], Feng [9], and Sukkar [33] have investigated alternative features, including durations (state, unit, word, etc.) and acoustic features (e.g., state's average cepstrum). Durational, acoustic, and language knowledge have been particularly useful in large vocabulary speech recognition systems [25], [36]. Other issues in keyword verification, such as reducing the amount of task-specific speech training data and the use of context-dependent acoustic models, have been addressed by Rose and Hofstetter [28], [29].

To improve the discrimination between keywords and out of vocabulary speech, several studies have introduced discriminative training techniques following maximum likelihood estimation. Rose [26] applied maximum mutual information (MMI) [1] estimation for the discrimination of keywords from a broad class of acoustic events. Villarrubia and Acero [35] applied affine transformation to the log-probability of the filler model. Chang and Lippmann [4] applied a figure of merit[1] backpropagation training, which eliminated the use of

Manuscript received June 3, 1995; revised July 22, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kuldip K. Paliwal.

M. G. Rahim is with AT&T Research Laboratories, Murray Hill, NJ 07974-0636 USA (e-mail: mazin@research.att.com).

C.-H. Lee and B.-H. Juang are with the Speech Research Department, Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: bhj@research.bell-labs.com).

Publisher Item Identifier S 1063-6676(97)03186-6.

[1] Figure of merit is the average detection rate over one to ten false alarms per keyword per hour.

thresholds during the training process. Other studies included performing some type of linear transformation or discriminative feature analysis [31]–[33].

As a generalization to keyword verification, *utterance verification* attempts to reject or accept part or all of an utterance based on a computed confidence score. It also attempts to reject erroneous but valid keyword strings (the so-called putative errors). This is particularly useful in situations where utterances are spoken without valid keywords, or when significant confusion exists among keywords, which may result in a high substitution error probability. In general, to deal with these types of problems, recognizers must be equipped with both a keyword spotting capability to correctly recognize keywords embedded in extraneous speech, and with an utterance verification capability to reject utterances that do no contain valid keywords and utterances that have low confidence scores.

In a study on utterance verification, Sukkar and Wilpon [33] introduced a two-stage method where the likelihood scores as well as the scores of a segmental generalized probabilistic descent (GPD) method [6] were combined using linear discriminant analysis to provide a keyword/nonkeyword decision. Significant improvement over a HMM-based classifier was reported when the two-stage approach was applied to operator-assisted calls. Good performance was later reported when a similar approach was applied for verification of connected digits strings [32].

This paper describes a HMM-based recognition/verification system. A two-pass strategy, with recognition followed by verification, is adopted. In the first pass, recognition is performed via a conventional Viterbi beam search algorithm, which segments the test utterance into a string of keyword hypotheses or N-best strings of keyword hypotheses. In the second pass, utterance verification is performed, which computes a confidence measure that determines whether or not to reject the recognized strings. Utterance verification is formulated as a statistical hypothesis test where the task is to test the *null* hypothesis that a given keyword or a set of keywords exists within a segment of speech against the *alternative* hypothesis that such keyword or keyword set does not exist, or is incorrectly classified, within that speech segment. Based on the well known Neyman–Pearson Lemma [2], a verification test can then be constructed using a likelihood ratio statistic. In real operational cases involving HMM systems, however, neither the null nor the alternative hypotheses can be evaluated exactly. It also complicates the issue that some type of *composite* alternative hypothesis is needed to provide improved discrimination between keywords and out-of-vocabulary words as well as improved detection of near-misses in keyword recognition. To facilitate this capability, this study will investigate the use of two sets of models to represent the alternative hypothesis, namely, *antikeywords* and *filler*. Considering that this test is not guaranteed to be optimal for HMM-based recognition, we will investigate the use of discriminative hypothesis testing where a class of discriminant functions is used to perform classification and hypothesis testing, and the required parameters are discriminatively trained using the available training data. One such class based on minimum classification error (MCE) training objective and the GPD training algorithm will be discussed in detail and later evaluated on a speaker-independent connected digits task.

The rest of the paper is organized as follows. Section II discusses statistical hypothesis testing for utterance verification, and motivates the use of the discriminative training methodology. Section III presents different strategies for reporting verification results. Section IV describes the database used in our experiments as well as the front-end process of our recognition system. Section V reviews the training/recognition/verification system. Section VI presents several formulations for digit verification based on likelihood scores of three types of models, namely, keywords, antikeywords, and filler. The construction of a string verification score based on the combined verification scores of the digits is discussed in Section VII. Section VIII outlines the use of discriminative minimum error training in HMM's and presents experimental results when applying MCE/GPD training to utterance verification. Section IX discusses a number of open issues in utterance verification and outlines directions for future efforts. Finally, a summary and conclusions will be given in Section X.

II. Statistical Hypothesis Testing

For a given speech segment $O = \{O_1, O_2, \cdots, O_t\}$, the purpose of pattern classification is to determine to which class $C(O) \in \{C_k; k = 1, \cdots, K\}$ the segment belongs. If the conditional probability $p(O \mid C_k)$ and the *a priori* probability $p(C_k)$ are assumed known, then the optimal class decision $\hat{C}(O)$ that minimizes the classification error is the Bayes decision rule that maximizes the *a posteriori* probability such that

$$\hat{C}(O) = \arg\max_k p(C_k \mid O) = \arg\max_k p(O \mid C_k)p(C_k). \tag{1}$$

On the other hand, for statistical hypothesis testing, the problem formulation is to test the *null hypothesis*, H_0, that a given keyword, KW_k, exists and is correctly recognized within a segment of speech, O, against the alternative hypothesis, H_1, that KW_k does not exist, or is incorrectly classified, within that speech segment. If again the probabilities of the null and the alternative hypotheses are known exactly, then according to the Neyman–Pearson Lemma [2], the optimal test (in the sense of maximizing the power of the test) is usually the probability ratio test such that the null hypothesis, H_0, is accepted if

$$\mathcal{LR}(k) = \frac{p_k(O \mid H_0)}{p_k(O \mid H_1)} > \tau_k. \tag{2}$$

This is referred to as the *likelihood ratio test* where $p_k(O \mid H_0)$ and $p_k(O \mid H_1)$ are the probability density functions (pdf's) of the null and the alternative hypotheses, respectively, and τ_k is the *critical threshold* of the test [2], [10].[2]

For testing simple hypotheses where the pdf's of H_0 and H_1 are known exactly, the likelihood ratio test is often the most powerful test for a given level of significance. For HMM-based speech recognition/classification systems, H_0 represents the class C_k which is either a sound, a subword unit, a whole-word

[2]The terms *critical threshold* and *verification threshold* are both used interchangeably in this paper.

unit, or even an utterance. H_1, on the other hand, represents other classes $\{C_j\}$, s.t. $j \neq k$. The parameters λ_k of the class C_k typically represent the state transition matrix, the state observation probability and the initial state probability. In this framework, however, both $p_k(O \mid H_0)$ and $p_k(O \mid H_1)$ can only be estimated by assuming a parametric form of the conditional densities and the distribution of the hypotheses. Clearly, any assumption of a parametric distribution may cause a mismatch between the "true" and estimated conditional distributions. This possible mismatch as well as a possible estimation error due to insufficient training data invalidate the optimality of the Bayes decision rule and the likelihood ratio test implied by the Neyman-Pearson Lemma.

In an effort to alleviate some of these problems, discriminative hypothesis testing is sought where a class of discriminant functions is used to perform classification and hypothesis testing. The form of the discriminant functions is required to be specified and their parameters are estimated from the training data. One such class of discriminant functions, based on MCE and GPD training, is described in Section VIII. A two-class problem is formulated where the null hypothesis assumes that the test utterance O is correctly recognized as class C_k and the alternative hypothesis assumes that O is incorrectly recognized as class C_k. Based on this framework, we define a *class confidence measure* (or *discriminant function*), $g_k(O;\Lambda)$, for class C_k ,which evaluates the confidence of accepting the null hypothesis that $O \in C_k$. The parameter set, Λ, in $g_k(O;\Lambda)$ is the recognition model parameters. We also define an *anti-discriminant measure* $G_k(O;\Lambda)$ which is used to evaluate how "unlikely" O contains C_k. A measure similar to the log likelihood ratio in (2) can now be defined as a function of the difference between $g_k(O;\Lambda)$ and $G_k(O;\Lambda)$

$$d_k(O;\Lambda) = -g_k(O;\Lambda) + G_k(O;\Lambda). \tag{3}$$

$d_k(O;\Lambda)$ is referred to as the *misclassification* measure. The MCE/GPD training algorithm involves finding a set of parameters for each class that minimizes the expected value of $d_k(O;\Lambda)$. The implied discriminative test is equivalent to maximizing the likelihood ratio, $\mathcal{LR}(k)$, since

$$\mathcal{LR}(k) = -d_k(O;\Lambda). \tag{4}$$

This is similar in spirit to the normalized verification function defined in [15] for speaker verification. In this paper, the MCE/GPD method will be used for training the filler and the keyword (i.e., recognition) models. The effect of applying discriminative training of this nature on both recognition and verification will be discussed.

III. Performance Evaluation

Statistical hypothesis testing is often evaluated based on two types of error measurements, namely, false rejection (or Type I) and false acceptance (or Type II—also referred to as false alarm). The former type of error occurs when a null hypothesis (e.g., keyword) is rejected, whereas the latter type occurs when a nonvalid null hypothesis (e.g., nonkeyword), or a valid keyword that is incorrectly recognized, is accepted. By appropriate selection of the critical threshold τ_k, it is possible

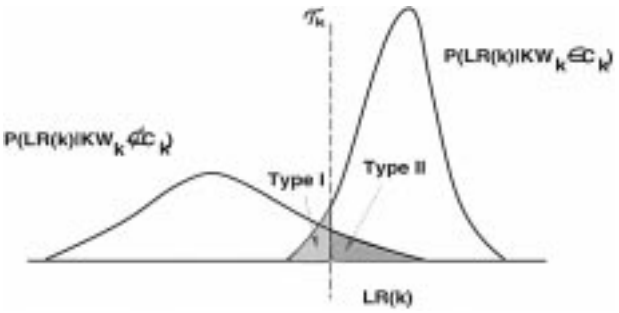

Fig. 1. Example showing histograms of the likelihood ratio $\mathcal{LR}(k)$ when $KW_k \in C_k$ and $KW_k \notin C_k$.

to provide a trade-off between Type I and Type II errors. For example, one may choose to have an *equal error rate*, which would require setting τ_k to provide equal amounts of Type I and Type II errors. Alternatively, τ_k may be chosen to provide a *minimum total error rate* of Type I plus Type II errors. Both of these measurements will be frequently utilized in this paper.

To select an appropriate operating point in utterance verification, it is conventional to plot a histogram of $\mathcal{LR}(k)$ for all training samples from class C_k and another histogram for all training samples *not* from class C_k. An example of such a representation is shown in Fig. 1. This representation serves as a way to quantify approximate Type I and Type II errors. The shaded area to the right of τ_k represents the amount of Type II error and the shaded area to the left of τ_k represents the amount of Type I error. The dashed line corresponds to the point where $p(\mathcal{LR}(k) \mid KW_k \in C_k) = \tau_k p(\mathcal{LR}(k) \mid KW_k \notin C_k)$.

To provide a more accurate representation of the errors, a receiver operating characteristic (ROC) curve may be used. An example is shown in Fig. 2, which displays the detection rate (1—Type I) versus the false alarm rate (Type II) as the operating point is varied. This type of representation has two benefits. First, the diagonal line from the top left to the bottom right corners of the plot intersects the ROC curve at the equal error rate point. Second, the performance curve provides an overall picture of the trade-off between Type I and Type II errors when varying the operating point. This helps in selecting an appropriate operating point to satisfy a particular application requirement. Notice that fewer errors incur when the performance curve approaches the top left corner point of the plot.

Due to the fact that ROC curves do not provide the necessary tool to compare different statistical tests, since they possess no information that relates τ_k to the various error levels, it is sometimes necessary to have yet another representation to show the error amount as a function of τ_k. An example of such a representation is shown in Fig. 3. The two dotted curves represent the accumulated Type I and Type II errors. The solid curve represents the total error rate. From this plot, one could deduce the minimum total error rate as well as the equal error rate. The use of histograms, ROC's and error rate curves will be frequently seen in this paper.

A final thought regarding performance evaluation for keyword spotting and utterance verification is that one should distinguish between making an error due to out-of-vocabulary words and an error due to a confusion among valid keywords.

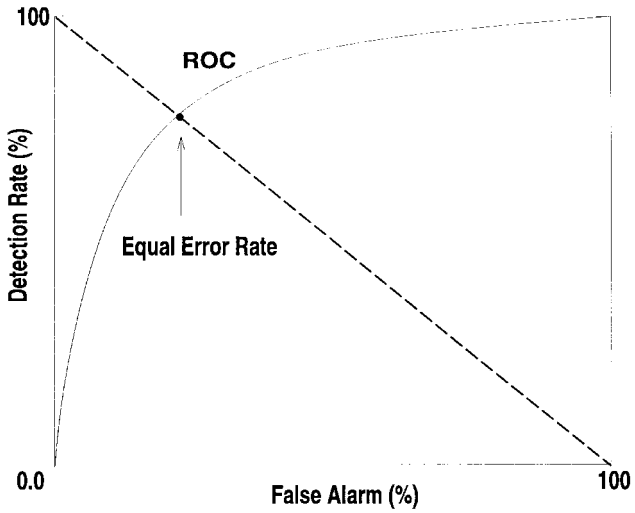

Fig. 2. Example of a ROC performance curve.

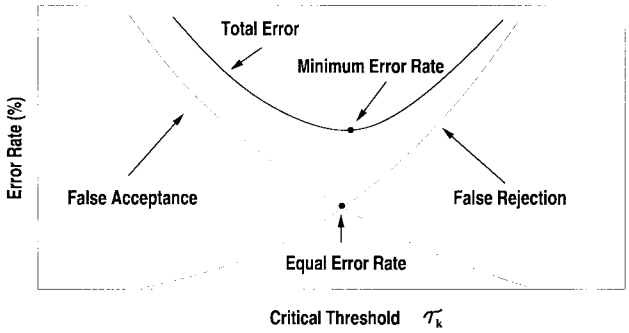

Fig. 3. Example showing the variations of Type I, Type II, and total errors as a function of τ_k.

For example, a keyword KW_k may be detected as a valid keyword but identified from class $C_j|_{j\neq k}$. In this paper, such errors will be referred to as "putative" errors. Putative errors can also be of Type I and Type II. When performing utterance verification, two levels of verification will be carried out. The first tests whether a valid digit string is being detected. The second tests for any putative errors. These two tests could take advantage of different statistical information, however, only one set of statistics is utilized in this study, as will be discussed in Sections VI and VII.

IV. Speech Database and Feature Extraction

A speaker-independent telephone-based connected digits database was used in this study to evaluate the verification system. This database was collected across two regions, namely, Long Island, NY, and Boston, MA, over a digital T1 interface. Speech was recorded using four different microphone handsets, two electret and two carbon button. Digit strings of lengths 10, 14, 15, and 16 digits, corresponding to credit card numbers and long-distance telephone numbers, were collected from 250 adult talkers (125 males and 125 females). Approximately half of the speakers were used for training the HMM's and the other half for testing and evaluating the various verification techniques. The training set and the testing set consisted of 2735 and 2842 valid-digit strings, respectively.

In order to provide nonkeyword utterances for training and verification, about 6000 phonetically rich sentences, modeled after the TIMIT sentences [14], was collected using the same recording environment as before. Half of this data was applied for training the filler model and the other half was used for testing.

The front-end feature extraction process was conducted as follows. Input speech, sampled at 8 kHz, was initially preemphasized (1-0.95z^{-1}) and grouped into frames of 240 samples with a shift of 80 samples. For each frame, a Hamming window was applied followed by autocorrelation analysis and LPC analysis using a tenth-order Durbin's recursion method [18]. A 12-dimensional LPC-derived cepstral vector was then computed and liftered using a weighting of the form

$$W_c(m) = \left[1 + 6\sin\left(\frac{\pi m}{12}\right)\right], \qquad 1 \leq m \leq 12. \tag{5}$$

The first- and second-time derivatives of the cepstrum, the so-called delta-cepstrum and delta-delta cepstrum, were also computed. Besides the cepstral-based features, the log of the energy, normalized by the peak sample, and its first- and second-order time derivatives, were also computed. Thus, each speech frame was represented by a vector of 39 features consisting of 12 cepstrum, 12 delta-cepstrum, 12 delta-delta cepstrum, 1 energy, 1 delta-energy and 1 delta-delta energy. Following feature extraction, signal bias removal was applied for channel normalization [20], [21].

V. Training/Recognition/Verification System

Each keyword (i.e., digit) is modeled by an N_{st}-state continuous density left-to-right HMM with M_{mix} mixture Gaussian state observation. The PDF for the observation vector O_t from state S_j and HMM λ_k is defined as

$$p(O_t \mid S_j, \lambda_k) = \sum_{m=1}^{M_{\text{mix}}} c_{kjm} \mathcal{N}(O_t; \mu_{kjm}, \Sigma_{kjm}) \tag{6}$$

where c_{kjm} is the mixture weight and $\mathcal{N}()$ is a multivariate Gaussian distribution with mean μ_{kjm} and diagonal covariance Σ_{kjm}.

Training of each keyword model consisted of estimating the mean, covariance, and mixture weights for each state using maximum likelihood (ML) estimation (e.g., [17]). In this study, the state transition probabilities were fixed at 0.5. For each keyword model, an antikeyword model was also trained. An antikeyword can be considered as a digit-specific filler model. It is based on a similar concept to the cohorts in speaker verification [30]. An antikeyword model $\overline{\lambda}_k$ is generally trained on the data of all keywords *but* that of keyword KW_k. Further explanation of this will be provided in the next section.

Aside from keywords and antikeywords, we also introduced a general acoustic filler model trained on nondigit speech data, and a background/silence model trained on the nonspeech segments of the signal. Therefore, a total of 24 models were used, corresponding to 11 keywords, 11 antikeywords, filler, and background/silence. Each model was represented by a ten-state HMM with 16 Gaussian densities per state, with the

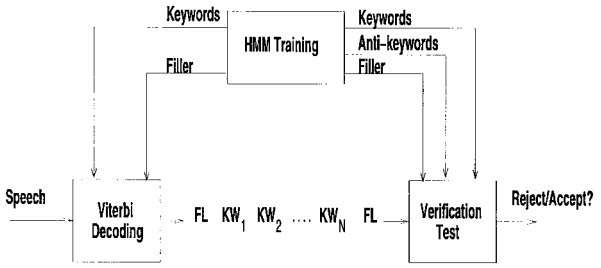

Fig. 4. Block diagram of the training/recognition/verification system.

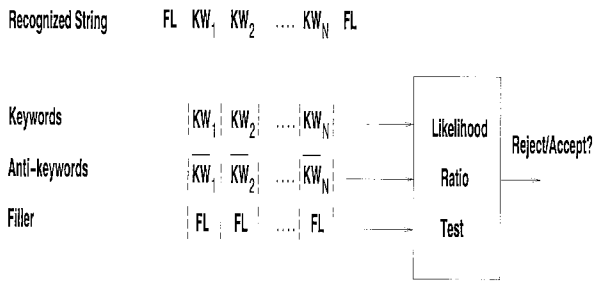

Fig. 5. Schematic diagram of the verification test.

exception of the background/silence that included a single state of 32 Gaussian densities.

A block diagram of the training, recognition, and verification system is shown in Fig. 4. A two-pass strategy is adopted consisting of recognition followed by verification. In the first pass, recognition is performed via a conventional Viterbi beam search algorithm, which segments the test utterance into a string of keyword hypotheses. In the second pass, an utterance-based confidence measure is constructed by combining the likelihood scores of all keywords and their corresponding antikeyword and filler models. In the example shown in Fig. 5, each antikeyword model $\overline{KW}_k$ is specific to keyword KW_k while the filler model, FL, is the same for all keywords. A likelihood ratio test is then performed, and the utterance as a whole is either accepted or rejected. Further details are presented in Section VII.

VI. Digit-Based Verification

Starting with four sets of HMM's, namely, 11 digits $\{\lambda_k\}$, 11 digit-specific antidigits $\{\bar{\lambda}_k\}$, silence/background λ_s and filler λ_f, digit verification is carried out by testing the null hypothesis that a specific digit exists in a segment of speech O versus the alternative hypothesis that the digit is not present. Based on the likelihood ratio test given in (2), the digit is accepted or rejected if its likelihood ratio $\mathcal{LR}_k(O \mid \Lambda)$ lies above a specific verification threshold τ_k (here, $\Lambda = \{\lambda_k\}, \{\bar{\lambda}_k\}, \lambda_s, \lambda_f$).

In this study, we considered four different formulations for the alternative hypothesis [i.e., $p_k(O \mid H_1)$ in (2)]. The first choice is simply to use the general acoustic filler model λ_f, which is *digit independent*. This is trained using nondigit extraneous speech and is the same for all digits. The likelihood for the alternative hypothesis is defined as

$$G_k^{(1)}(O;\Lambda) = \frac{1}{T_k}\log[p(O \mid \lambda_f)] \tag{7}$$

where T_k is the number of frames allocated for digit k. This type of formulation is believed to improve discrimination between keywords and out of vocabulary words.

The next two choices for the alternative hypothesis introduce a digit-specific antidigit model in order to provide better detection of near misses in digit recognition. We first considered using a type of a geometric mean of all competing digit models. For digit model λ_k, for example, the corresponding antidigit function would be

$$G_k^{(2)}(O;\Lambda) = \log\left[\frac{1}{N-1}\sum_{j,j\neq k}\exp\{\kappa g_j(O \mid \Lambda)\}\right]^{\frac{1}{\kappa}} \tag{8}$$

where N is the total number of digit models (i.e., 11), κ is a positive constant, and $g_j(O \mid \Lambda)) = \frac{1}{T_j}\log[p(O \mid \lambda_j)]$. This type of discrimination is believed to improve detection of near misses in digit recognition. Therefore, digit strings with possible putative errors could be detected. One would notice that the alternative hypothesis in (8) is somewhat similar to the concept of cohorts in speaker recognition [30]. It is also similar to the antidiscriminant function defined in minimum error discriminative training (see Section VIII) [6], [7], [12], [15]. In this formulation, if κ is set to infinity, then only the first competing digit (i.e., second best) would be considered.

The obvious problem with the antidigit function in (8) is that N-best digit scores would be needed for each digit hypothesis in order to compute the geometric average. If computational cost is an issue, then this type of formulation would pose a problem. To obtain a valid approximation of the same function without having to compute likelihood of all competing digits, we have trained 11 digit-specific antidigit models $\{\bar{\lambda}_k\}$ using the same ML training procedure for obtaining the digit models. Each model, $\bar{\lambda}_k$, is trained on all digits except of the data for digit k

$$G_k^{(3)}(O;\Lambda) = \frac{1}{T_k}\log[p(O \mid \bar{\lambda}_k)]. \tag{9}$$

By using the function in (9), an antidigit score becomes available without having to compute a word N-best hypothesis but at the expense of increasing the number of models. Therefore, there is a choice of either more computation when applying $G_k^{(2)}(O;\Lambda)$, or more memory when applying $G_k^{(3)}(O;\Lambda)$.

The next choice of an antidigit function is to combine both $G_k^{(1)}(O;\Lambda)$ and the best of $G_k^{(2)}(O;\Lambda)$ or $G_k^{(3)}(O;\Lambda)$, so that to achieve improved discrimination between keyword and nonkeyword models as well as reasonable detection of putative errors. Although there are many ways for constructing such a function, a simple average was chosen thus defining the alternative hypothesis for digit k as

$$\begin{aligned} G_k^{(4)}(O;\Lambda) &= \log\left[\gamma\cdot\exp\left\{\kappa G_k^{(1)}(O;\Lambda)\right\}\right. \\ &\qquad \left. + (1-\gamma)\cdot\exp\left\{\kappa G_k^{(2)}(O;\Lambda)\right\}\right]^{\frac{1}{\kappa}} \\ \text{or} \qquad & \qquad (10) \\ G_k^{(4)}(O;\Lambda) &= \log\left[\gamma\cdot\exp\left\{\kappa G_k^{(1)}(O;\Lambda)\right\}\right. \\ &\qquad \left. + (1-\gamma)\exp\left\{\kappa G_k^{(3)}(O;\Lambda)\right\}\right]^{\frac{1}{\kappa}} \end{aligned}$$

where γ is a weighting that has been set to 0.5 in this study.

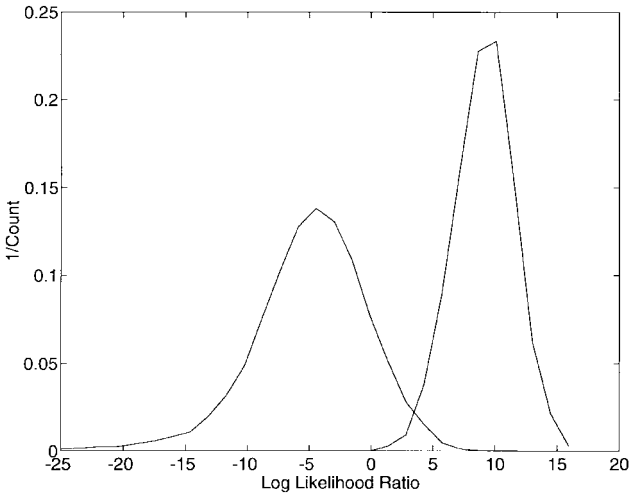

Fig. 6. Histograms for digit 9 using a likelihood ratio distance based on $G_9^{(1)}(O;\Lambda)$.

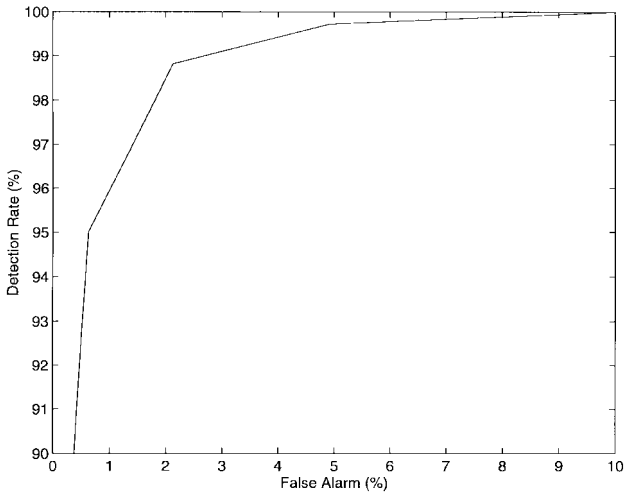

Fig. 7. ROC performance curve for digit 9 using a likelihood ratio distance based on $G_9^{(1)}(O;\Lambda)$.

In order to compare the above four criteria for representing the alternative hypothesis, one can construct two histograms for each keyword consisting of in-class and out-class likelihood ratio scores (as suggested in Fig. 1). The overlap between each two histograms corresponds to the amount of confusion of the keyword with all other keywords and nonvocabulary words. We will consider the digit "9" as an example. Fig. 6 shows the two histograms computed using $G_9^{(1)}(O;\Lambda)$. The histogram on the right represents the distribution of the training samples from the digit 9. Similarly, the histogram on the left represents the distribution of the training samples that are not from the digit 9 (i.e., this involves the rest of the digits as well as the nonvocabulary words). From this representation, approximate Type I and Type II errors can be computed for each digit for a given choice of τ_k (see Fig. 1).

By changing the value of τ_k for digit 9, a ROC curve can be constructed as shown in Fig. 7. This representation is useful since it provides an overall picture of the amounts of error that would be incurred when operating at different verification thresholds. For example, the equal error rate point for the digit 9 is 1.9% when setting τ_k to 3.1. The minimum total error rate is 3.3% when setting τ_k to 2.8.[3] If a new operating point, say 4.3, is established, then the Type I and Type 2 errors become, 5.0 and 0.7, respectively. Ideally, one would like to have a minimal change in the error rate when varying the operating point slightly. This touches the issue of robustness, which is addressed in [19] and [23].

With the aid of histograms and ROC curves, we now need to determine which of the antidigit functions presented in (7) to (11) is best suited for digit verification. A series of experiments was performed. The first experiment compared $G_k^{(2)}(O;\Lambda)$ and $G_k^{(3)}(O;\Lambda)$. Recall that both measures incorporate the concept of digit-specific antidigit model with the former requiring the computation of an N-best digit hypothesis (N was set to 4) and the latter requiring the construction of an antidigit HMM. Fig. 8 shows the ROC curves of the two functions for the digit

[3]Note that different operating points may be needed to achieve equal error rate and minimum total error rate.

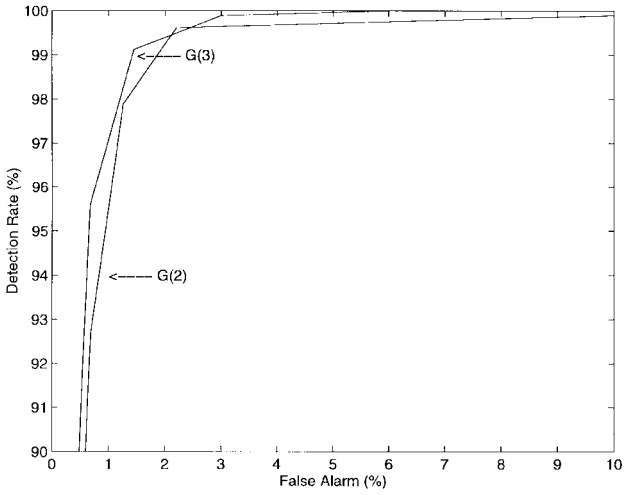

Fig. 8. ROC performance curves for digit "9" using likelihood ratio distances based on $G_9^{(2)}(O;\Lambda)$ and $G_9^{(3)}(O;\Lambda)$.

9. Although the curves are quite comparable, using a likelihood ratio score based on $G_k^{(3)}(O;\Lambda)$ versus $G_k^{(2)}(O;\Lambda)$ results in a smaller error rate. A similar observation was noticed when we evaluated on the remaining set of digits. For this reason, we opted to use $G_k^{(3)}(O;\Lambda)$, which will avoid the additional computational effort needed to perform word N-best but at the expense of nearly doubling the number of models.

The second experiment was performed to verify the role of introducing a digit-specific antidigit model. We evaluated the two functions $G_k^{(1)}(O;\Lambda)$ and $G_k^{(4)}(O;\Lambda)$ since the former considers a filler model only and the latter uses both the filler and the antidigit function of $G_k^{(3)}(O;\Lambda)$. Upon examination of the histograms of these two functions in Fig. 9, it appears that $G_9^{(4)}(O;\Lambda)$ is able to provide a better class separation than $G_9^{(1)}(O;\Lambda)$, resulting in lower Type I and Type II errors. The two corresponding ROC curves for the two functions are shown in Fig. 10. Again, the advantages of representing the alternative model by an antikeyword, versus the use of a filler model alone, is clearly demonstrated.

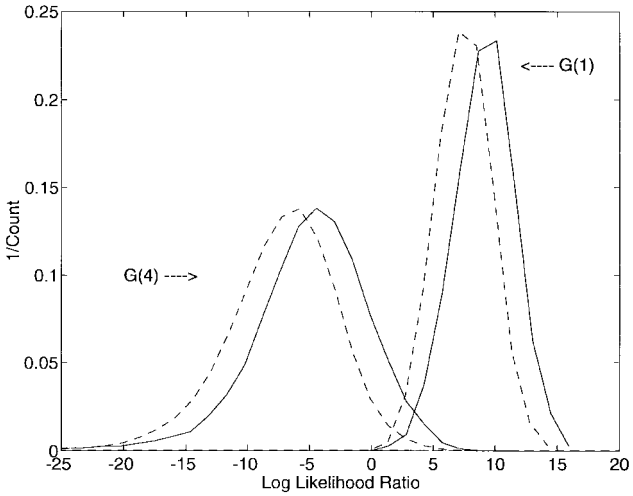

Fig. 9. Histograms for digit "9" using likelihood ratio distances based on $G_9^{(1)}(O;\Lambda)$ and $G_9^{(4)}(O;\Lambda)$.

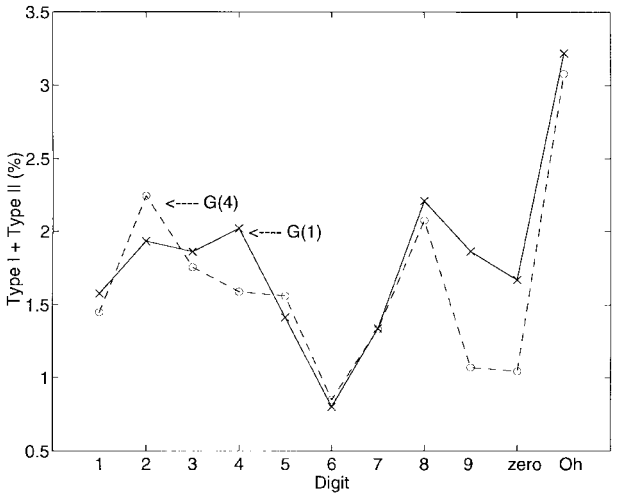

Fig. 11. Equal error rates for all the digits using likelihood ratio distances based on $G_k^{(1)}(O;\Lambda)$ and $G_k^{(4)}(O;\Lambda)$.

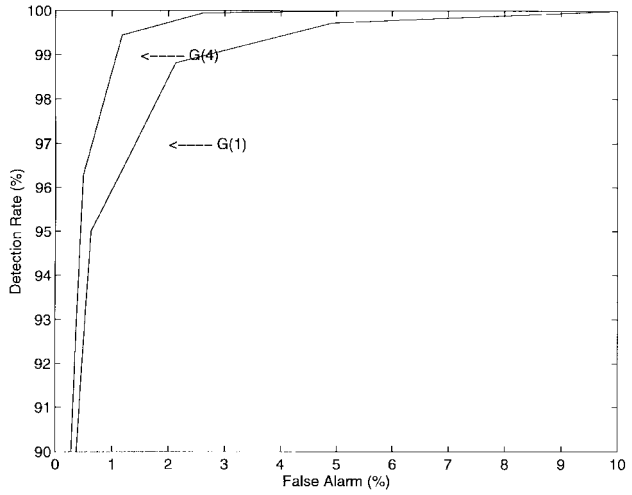

Fig. 10. ROC performance curves for digit "9" using likelihood ratio distances based on $G_9^{(1)}(O;\Lambda)$ and $G_9^{(4)}(O;\Lambda)$.

Fig. 11 gives the equal error rates for all the eleven digits when utilizing a likelihood ratio distance based on either $G_k^{(4)}(O;\Lambda)$ or $G_k^{(1)}(O;\Lambda)$. Clearly, for almost all digits, it is safe to conclude that digit-specific antidigits are somewhat complementary to a general acoustic filler model. Combining the scores of both sets of models in a geometric average has resulted in an improved performance and consequently lower both the Type I and Type II errors. A similar trend was found when plotting the minimum total error rate for all digits. It should be noted that these results are substantially better than those obtained when using absolute likelihood scores only (i.e., with no alternative hypothesis). The equal error rate when using absolute likelihood scores, averaged over all digits, was about 6.5% higher in value than the results shown in Fig. 11.

Throughout the rest of the paper, digit verification will be conducted using a likelihood ratio test with the antidigit function $G_k^{(4)}(O;\Lambda)$.[4] Since the objective of this study is to perform utterance verification rather than keyword rejection, it is essential to define an utterance-based likelihood measure. This problem will be dealt with in the next section.

[4]For simplicity of notation, $G_k^{(4)}(O;\Lambda)$ will be written as $G_k(O;\Lambda)$ for the remainder of the paper.

VII. Utterance-Based Verification

There are several advantages in using utterance verification (or rejection) in connected digits recognition. The first is verifying whether the recognized digit string is a *valid* digit string. This enables rejection of strings that contain nonvocabulary words or noise. The second is verifying whether a valid digit string is a *correct* digit string. This is a more difficult task than the previous one, since it is dependent on the reliability of the recognizer. Detecting incorrectly recognized digit strings improves the performance and the usability of the recognition system. The third determines which parts of the valid digit string is reliable. The system may prompt the user to provide only the part of speech that is unreliable, for example, "please repeat the first three digits." Therefore, different string-level tests need to be investigated.

In this study, we do not consider verification of partial information. A digit string is either totally accepted or rejected based on its confidence score. Two approaches for computing the confidence score have been investigated. In the first approach, the utterance confidence measure relies on individual digit scores, such that an utterance is rejected if the test on any detected digit q

$$S^{(1)}(O;\Lambda) = \mathcal{LR}_q(O;\Lambda) < \tau_q \tag{11}$$

where

$$\mathcal{LR}_q(O;\Lambda) = g_q(O;\Lambda) - G_q(O;\Lambda) \tag{12}$$

(i.e., reject if any one detected digit falls below a critical threshold, τ_q). This measure can be relaxed by allowing a string to be rejected only if the likelihood ratio scores of multiple digits fall below a specified critical threshold.

The second approach for utterance verification computes a string-based confidence measure by averaging the likelihood scores of all detected digits. Thus, for a J-digit string, we

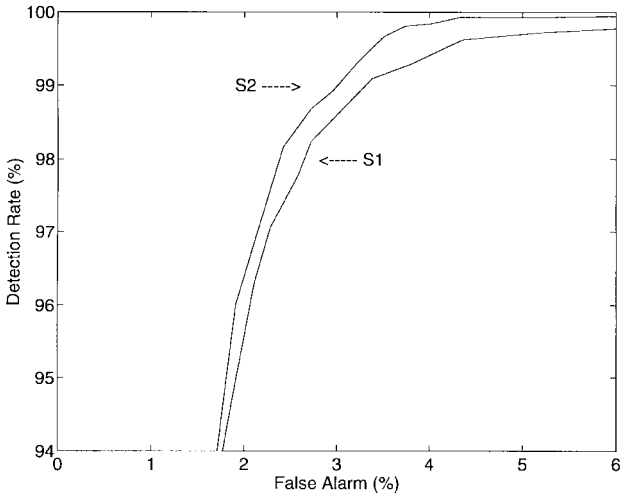

Fig. 12. ROC performance curves for $S^{(1)}(O;\Lambda)$ and $S^{(2)}(O;\Lambda)$.

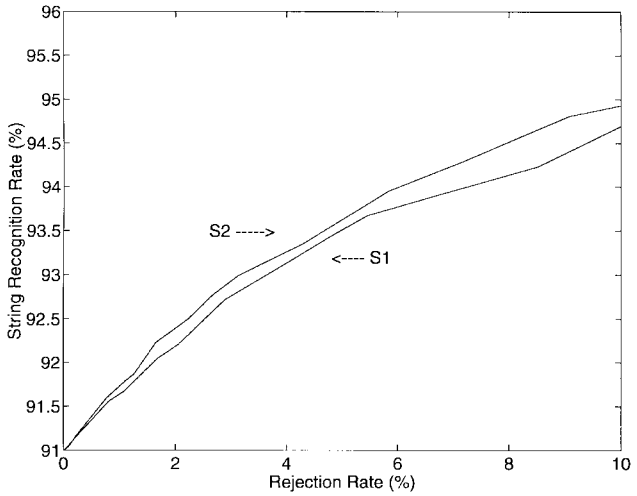

Fig. 13. String recognition performance as a function of rejection rate.

have the following:

$$S^{(2)}(O;\Lambda) = -\log\left[\frac{1}{J}\sum_{q=1}^{J}\exp\{-\eta\cdot\mathcal{LR}_q(O;\Lambda)\}\right]^{\frac{1}{\eta}} \quad (13)$$

where η is a positive constant. There are two advantages of using this measure compared to $S^{(1)}(O;\Lambda)$. First, it provides string verification statistics based only on one distribution rather than one per digit. Second, this measure weights the contributions of all the digits within a given string based on the selected value of η (note that η can be digit specific). This is believed to be important since digit strings with multiple putative errors would be more easily detected with this type of formulation. Also, a putative error causes almost all neighboring segmentations to be changed and consequently affecting $S^{(2)}(O;\Lambda)$ more than $S^{(1)}(O;\Lambda)$. If $\eta \gg 1$ then $S^{(2)}(O;\Lambda) \simeq \min_q \mathcal{LR}_q(O;\Lambda)$ (i.e., the lowest score). Currently, η is set to unity.

The two utterance verification functions defined in (11) and (13) have been evaluated on the testing database. Let us first consider a combined error measure including both nonvocabulary words and putative errors. By varying the critical threshold for each function, a ROC performance curve is obtained as shown in Fig. 12. This figure demonstrates that $S^{(2)}(O;\Lambda)$ achieves a lower error rate than $S^{(1)}(O;\Lambda)$ at all threshold values. The equal error rates for $S^{(1)}(O;\Lambda)$ and $S^{(2)}(O;\Lambda)$ are at 2.5% and 2.3%, respectively. This amounts to a reduction of 8%.

Let's now consider putative errors only. Assuming that nonvocabulary words never existed then the baseline string recognition performance of the system with *no* rejection is 91.0%. If utterance verification is to be employed using either $S^{(1)}(O;\Lambda)$ or $S^{(2)}(O;\Lambda)$, then it would be expected for the string recognition performance to improve when raising the rejection rate. Fig. 13 shows this exact behavior. Note that the rejection rate refers to the total rejection of correct digit strings and putative errors. The string recognition performance refers to the accuracy on the remaining digit strings after rejection. For example, at a rejection rate of 5%, it is possible to improve the string accuracy from 91.0% to 93.5% using $S^{(1)}(O;\Lambda)$, and from 91.0% to 93.7% using $S^{(2)}(O;\Lambda)$. Although this difference in performance is not significant (about 0.2% at a rejection rate of 5%), it is sufficient to prefer $S^{(2)}(O;\Lambda)$ over $S^{(1)}(O;\Lambda)$. Furthermore, it is expected that a larger improvement using $S^{(2)}(O;\Lambda)$ can be achieved when selecting a more appropriate value for η in (13). This is currently under investigation.

TABLE I
PERFORMANCE RESULTS WHEN APPLYING EITHER $S^{(1)}(O;\Lambda)$ OR $S^{(2)}(O;\Lambda)$ FOR UTTERANCE VERIFICATION AT THE POINT OF MINIMAL ERROR RATE

Function	Before Rej. (%)	Rej. Rate (%)	After Rej. (%)	Rej. non-voc. (%)
$S^{(1)}(O;\Lambda)$	91.0	2.9	92.7	99.7
$S^{(2)}(O;\Lambda)$	91.0	3.1	93.0	99.9

Table I summarizes the results when using either functions for verification at the point of minimal error rate. This is achieved by selecting an operating point for each function to best minimize the combined Type I and Type II errors. With a suitable operating point, it is clear that rejection of nonvocabulary strings is not a problem with performance that exceeds 99% (see column 5).

Since $S^{(2)}(O;\Lambda)$ has consistently shown improved performance over $S^{(1)}(O;\Lambda)$, yet providing a single string likelihood distribution versus one per digit, this measure will be used in all remaining experiments.[5]

VIII. DISCRIMINATIVE TRAINING

In Section II, we showed that the Bayes decision rule and the likelihood ratio test are not guaranteed to be optimal when applied for verification of HMM-based speech recognition systems. To alleviate this problem, we motivated the use of discriminative hypothesis testing, where a class of discriminant function is used to perform classification and hypothesis testing. One such class of discriminant functions based on minimum classification error (MCE) and generalized probabilistic descent (GPD) is reported in this section. The technique of MCE/GPD, proposed by Katagiri *et al.* [13] and Juang and Katagiri [12], has been extensively used in the

[5]For simplicity of notation, $S^{(2)}(O;\Lambda)$ will be written as $S(O;\Lambda)$ for the remainder of the paper.

area of speech recognition [6], [7] and speaker recognition [15]. In this study, we investigate the effect of this training paradigm for both recognition and verification. The training objective function will be formulated at the *string* level in order to consider all three types of errors, namely, insertions, deletions, and substitutions.

Unlike ML estimation, which maximizes a likelihood function of a sequence of observations given a set of HMM's, in MCE/GPD the goal is to minimize the expected loss function

$$L(\Lambda) = E[l\{d_i(O;\Lambda)\}] \tag{14}$$

where $l\{\cdot\}$ is a smooth function which is typically set to a sigmoid, and $d_i(O;\Lambda)$ is a *misclassification* measure for string class i. A three-step procedure is applied for estimating the expected loss function $L(\Lambda)$. The first step formulates the misclassification distance for string i

$$d_i(O;\Lambda) = -g_i(O;\Lambda) + G_i(O;\Lambda). \tag{15}$$

The discriminant function, $g_i(O;\Lambda)$, for the correct class C_i is defined as $\frac{1}{T}\log p(O;\Lambda)$, and the antidiscriminant function, $G_i(O;\Lambda)$ is defined as

$$G_i(O;\Lambda) = \log\left[\frac{1}{M-1}\sum_{j,j\neq i}^{M}\exp\{\eta\cdot g_j(O;\Lambda)\}\right]^{\frac{1}{\eta}}, \quad \eta > 0. \tag{16}$$

$G_i(O;\Lambda)$ can be considered as some type of a geometric mean of the likelihoods of the competing classes to C_i. In the current study, $G_i(O;\Lambda)$ was estimated using a four-best string hypothesis decoder [8]. Thus, the number of competing classes, M, is three when the correct string is among the top four candidates and M is four, otherwise. Note that the distance in (15) is negative if O is correctly classified and positive otherwise.

The next step is to approximate the misclassification error count. This is achieved using a smooth and differentiable 0-1 sigmoid function of the form

$$l_i(O;\Lambda) = l(d_i(O;\Lambda)) = \frac{1}{1+\exp\{-\alpha d_i(O;\Lambda)+\beta\}} \tag{17}$$

where α and β are constants which control the slope and the shift of the smoothing function, respectively.

The third and final step in MCE/GPD training involves finding the set of parameters Λ that minimize the expected value of the loss function, i.e., $L(\Lambda)$, in (17). The parameter set Λ (i.e., mixture means, variances and gains) is updated at every iteration n according to

$$\Lambda_{n+1} = \Lambda_n - \epsilon_n V_n \nabla L(\Lambda_n), \quad \epsilon_n > 0 \tag{18}$$

where ϵ_n is a learning rate and V_n is a positive definite matrix. Details of the derivation of the HMM parameters using the MCE/GPD technique are available in [7] and [12]. Throughout all our experiments, MCE/GPD was only applied for training the filler and the keyword models. At the time of writing this paper the digit-specific antidigit models were not trained with this technique. Training of these models would require a somewhat different paradigm as shown in [22] and [34].

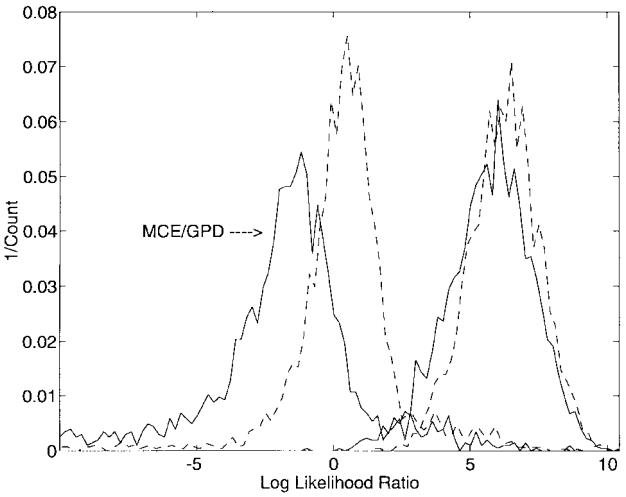

Fig. 14. Histograms showing the distribution of the string-based likelihood ratio scores, based on $S(O;\Lambda)$, before and after MCE/GPD training.

In the results reported by Liu *et al*. [15], it was shown that MCE/GPD training helped in pulling apart the in-class/out-class histograms of speaker verification scores, thus causing lesser Type I and Type II errors. This property was also observed in our study. Fig. 14 shows the two histograms for the in-class/out-class string likelihood scores, based on (13), when applying ML training (dotted lines) and following MCE/GPD training (solid line). Clearly, the discriminative training technique has provided a better separation of the two distributions, a feature that is more apparent in the left distribution representing the incorrect class.

Naturally, since the histograms are less overlapped than those previously obtained with ML training, a decrease in the error rate would be expected. Fig. 15 shows the variation of the total Type I and Type II errors for ML and MCE/GPD training when changing the critical threshold between -1 and 5. The total error rate plot following MCE/GPD training is shifted toward the origin and is clearly less sensitive to variations in the critical threshold. For example, to obtain less than a 10% total error rate for ML training, the critical threshold can be set anywhere between 1.5 and 5.2. For the same total error rate of 10%, the critical threshold following MCE/GPD training can be set anywhere between -0.7 and 4.3. The larger dynamic range in setting the critical threshold provides some degree of robustness to any possible acoustic mismatch between the training model and the testing data [19], [23].

Since the rejection rate of nonvocabulary word strings is in excess of 99%, it appears that the major challenge in utterance verification is the rejection of putative errors. When considering valid digit strings only, Fig. 16 shows the string recognition performance as a function of rejection rate. At a rejection rate of 5%, for example, the string recognition performance improves from 93.6% following ML training to 96.1% following MCE/GPD training. This corresponds to a

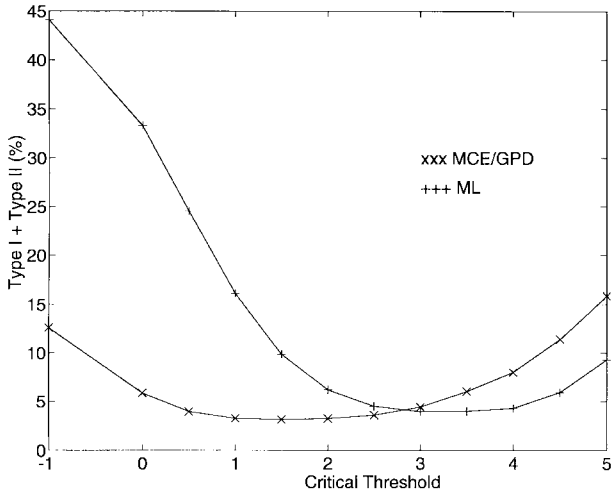

Fig. 15. Combined Type I and Type II errors with ML training and following MCE/GPD training.

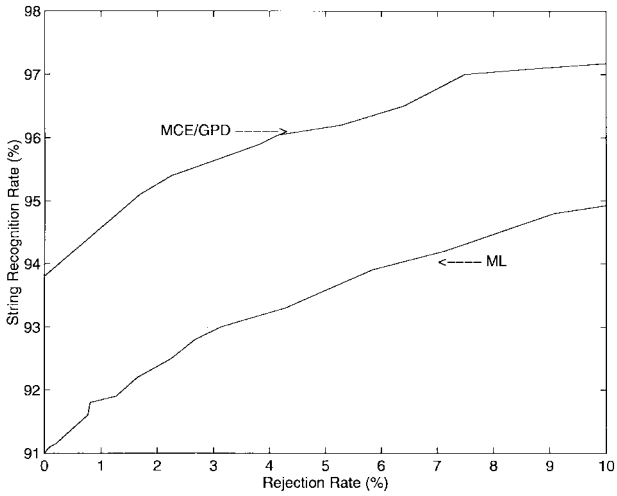

Fig. 16. String recognition performance as a function of rejection rate when introducing MCE/GPD training.

reduction in string error rate by about 39% which is consistent even up to a 10% rejection rate.

IX. Discussion

As the demand for speech recognition technologies increases, the need for the development of systems that are robust to speaking style, accents, environmental mismatch, disfluency, etc., is becoming increasingly essential. In these circumstances, utterance verification plays an important role in maintaining an acceptable error rate and in providing a desirable trade-off between false alarm rate and false rejection rate.

The work presented in this paper is a first step toward our vision of a *totally* robust utterance verification system. Robust verification is a subject that demands considerable attention. In a separate publication [19], [23], we show that utterances recorded under different environmental conditions require different operating points in order to satisfy a given optimality criterion. Experimentally, it is shown that the ROC statistics based on a particular training or evaluation set would not work optimally under mismatched testing conditions. Methods to alleviate this problem are reported in [19] and [23].

At present, we are investigating several different avenues to improving the verification performance. The first involves the development of a string-based likelihood ratio distance. Recall that the function, defined in (13), is basically a geometric average of digit likelihood ratio scores. Since the use of likelihood ratio distances, as opposed to likelihood distances, has resulted in a tremendous improvement in performance, it is believed that extending our formulation to include an antistring discriminant function would provide equal benefits.

Another avenue to minimizing false rejection errors and false alarms is to apply discriminative training to the digit-specific antidigit models. This would require a modification of the MCE formulation to accommodate for antikeyword models. Combining such an approach with MCE training in a two-pass strategy is expected to give a desired trade-off between a reduced Type I and Type II errors and a minimum string error rate [22], [34].

An effective approach to improving the performance of utterance verification systems is to introduce additional features, besides the likelihood scores, to help detect nonvocabulary words and putative errors more accurately. One example is to use state durational information that is known to be effective in detecting extraneous speech [32]. A different strategy to improving the verification performance is to use context-dependent subword units instead of the whole word models. From our experience, these types phonological units result in an improved performance in connected digits recognition. At the time of writing this paper, a verification system tailored toward subword units was under study [22], [34].

Finally, to provide a user-friendly speech recognition system, verification of partial information is essential. Users of a speech recognition system are typically impatient when being prompted to repeat their 16-digit credit card number, for example, more than once. Being asked to repeat a portion of the digit string is commonly more acceptable. Current study is focused on evaluating the success rate of the proposed verification system in identifying unreliable parts of a spoken digit string.

X. Summary and Conclusion

This paper presented an HMM-based system for connected digits recognition/verification. A two-pass strategy was adopted, consisting of recognition followed by verification. In the first pass, recognition was performed via a conventional Viterbi beam search algorithm. In the second pass, an utterance-based confidence score was computed and applied for verification.

For digit verification, we tested the null hypothesis that a specific digit exists in a segment of speech versus the alternative hypothesis that the digit was not present. Several formulations were investigated for the alternative hypothesis based on likelihood distances of digit-specific antidigit models and a general acoustic filler model. It was demonstrated that incorporating a geometric average that combined the scores of both sets of models resulted in reduced equal error rates.

TABLE II
STRING RECOGNITION PERFORMANCE AT DIFFERENT REJECTION RATES

Rej. Rate (%)	After ML (%)	After MCE/GPD (%)
0.0	91.0	93.8
1.0	91.8	94.5
3.0	92.9	95.6
5.0	93.6	96.1
7.0	94.1	96.8
9.0	94.8	97.1

For utterance verification, two approaches were investigated based on the likelihood ratio scores of digits. The first was to reject the digit string if the score of any detected digit falls below a specified digit-specific critical threshold. The second approach was to combine the likelihood scores of all detected digits using a type of a geometric average and then to reject the digit string if its confidence score falls below a specific string verification threshold. The latter approach was shown to give improved performance for connected digits as well as to provide a single string-based likelihood distribution as opposed to one distribution per digit. When evaluating the utterance verification system on a speaker-independent connected-digits database, the string error rate reduced by about 29% at 5% rejection rate. The string recognition performance at different rejection rates is shown in Table II. For rejection of nonvocabulary word strings, the proposed system rejected over 99.9% of the utterances.

In this paper, we illustrated that the Bayes decision rule and the likelihood ratio test are not guaranteed to be optimal when applied to verification of HMM-based speech recognition systems. To alleviate this problem, we investigated the use of discriminative hypothesis testing in the framework of minimum classification error training. A string-based MCE/GPD method was applied for training the filler and keyword models. Since the keyword models were used in both recognition and verification, it was established from our experimental results that MCE/GPD training helped not only to reduce the recognition error rate but also the verification error rate. Using this discriminative training method with a specific operating point, the string error rate was reduced by a further 39% at 5% rejection rate (see Table II). It was interesting to note that a similar reduction in error rate was also achieved at higher rejection rates.

In summary, the proposed utterance verification system rejected over 99.9% of nonvocabulary word strings and reduced the string error rate for valid digit strings by about 57% at 5% rejection. The application of this technique under mismatched environmental conditions is reported in [19] and [23].

ACKNOWLEDGMENT

The authors acknowledge useful discussions with W. Chou, A. Setlur, and R. Sukkar. We would also like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] L. Bahl, P. Brown, P. Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1986, vol. I, pp. 49–52.

[2] P. Bickel and K. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[3] H. Bourlard, B. D'Hoore, and J.-M. Boite, "Optimizing recognition and rejection performance in wordspotting systems," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1994, vol. I, pp. 373–376.

[4] E. Chang and R. Lippmann, "Figure of merit training for detection and spotting," in *Proc. Conf. Neural Information Processing Systems*, Denver, CO, 1993, vol. 6, pp. 1019–1026.

[5] B. Chigier, "Rejection and keyword spotting algorithms for a directory assistance city name recognition application," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1992, Vol. II, pp. 93–96.

[6] W. Chou, B.-H. Juang, and C.-H. Lee, "Segmental GPD training of HMM based speech recognizer," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1992, Vol. I, pp. 473–476.

[7] W. Chou, B.-H. Juang, C.-H. Lee, and F. K. Soong, "A minimum error rate pattern recognition approach to speech recognition," *J. Pattern Recog. Artif. Intell.*, vol. VIII, no. 1, pp. 5–31.

[8] W. Chou, T. Matsuoka, C.-H. Lee, and B.-H. Juang, "A high resolution N-best search algorithm using inter-word context dependent models for continuous speech recognition," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Adelaide, Australia, 1994.

[9] M.-W. Feng and B. Mazor, "Continuous word spotting for applications in telecommunications," in *Proc. Int. Conf. on Spoken Language Processing,*, 1992, pp. 21–24.

[10] K. Fukunaga, *Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.

[11] J. Godfrey, E. Holliman, and J. McDaniel, "SWITCH-BOARD: Telephone speech corpus for research development," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1992, vol. I, pp. 517–520.

[12] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Processing*, vol. 40, pp. 3043–3054, Dec. 1992.

[13] S. Katagiri, C.-H. Lee, and B.-H. Juang, "Discriminative multi-layer feed-forward networks," in *IEEE Proc. Neural Networks for Signal Processing*, 1991, pp. 11–20.

[14] L. F. Lamel, R. H. Kessel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proc. Speech Recognition Workshop (DARPA)*, 1986.

[15] C.-S. Liu, C.-H. Lee, W. Chou, B.-H. Juang, and A. Rosenberg, "A study on minimum error discriminative training for speaker recognition," *J. Acoust. Soc. Amer.*, accepted for publication.

[16] P. Moreno, D. Roe, and P. Ramesh, "Rejection techniques in continuous speech recognition using hidden Markov models," in *Proc. Europ. Conf. Signal Processing*, 1990, pp. 1383–1386.

[17] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[18] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ: Prentice-Hall, 1978.

[19] M. Rahim, C.-H. Lee, and B.-H. Juang, "Robust utterance verification for connected digits recognition," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing.*, 1995.

[20] M. Rahim, B.-H. Juang, W. Chou, and E. Buhrke, "Signal conditioning techniques for robust speech recognition," *IEEE Signal Processing Lett.*, vol. 3, 1996.

[21] M. Rahim and B.-H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 19–30, 1996.

[22] M. Rahim, C.-H. Lee, B.-H. Juang, and W. Chou, "Discriminative utterance verification using minimum string verification error (MSVE) training," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1996, vol. I.

[23] M. Rahim, C.-H. Lee, and B.-H. Juang, "A study on robust utterance verification for connected digits recognition," *J. Acoust. Soc. Amer.*, accepted for publication, May 1997.

[24] J. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker independent word spotting," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1989, pp. 627–630.

[25] J. Rohlicek, P. Jeanrenaud, K. Ng, H. Gish, B. Musicus, and M. Siu, "Phonetic training and language modeling for word spotting," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1993, vol. II, pp. 459–462.

[26] R. Rose, "Discriminant wordspotting techniques for rejecting nonvocabulary utterances in unconstrained speech," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1992, vol. II, pp. 105–108.

[27] R. Rose and D. Paul, "A hidden Markov model based keyword recognition system," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1990, vol. I, pp. 129–132.

[28] R. Rose and E. Hofstetter, "Techniques for robust wordspotting in continuous speech messages," in *Proc. Europ. Conf. Speech Communications*, 1991, pp. 1183–1186.
[29] _____, "Task independent wordspotting using decision tree based allophone clustering," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1993, vol. II, pp. 467–470.
[30] A. Rosenberg, J. Delong, C.-H. Lee, B.-H. Juang, and F. Soong, "The use of cohort normalized scores for speaker recognition," in *Proc. Int. Conf. Spoken Language Processing*, 1992, pp. 599–602.
[31] J. Sorensen and M. Savic, "Hierarchical pattern classification for high performance text-independent speaker verification systems," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 1994, vol. I, pp. 157–160.
[32] R. Sukkar, "Rejection for connected digit recognition based on GPD segmental discrimination," in *Proc. Int. Conf. Acoust., Speech, and Signal Processing*, 1994, vol. I, pp. 393–396.
[33] R. Sukkar and J. Wilpon, "A two pass classifier for utterance rejection in keyword spotting," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1993, vol. II, pp. 451–454.
[34] R. Sukkar, A. Setlur, M. Rahim, and C.-H. Lee, "Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1996, vol. I.
[35] L. Villarrubia and A. Acero, "Rejection techniques for digit recognition in telecommunication applications," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1993, vol. II, pp. 455–458.
[36] M. Weintraub, "Keyword-spotting using SRI's DECIPHER^{TM} large-vocabulary speech-recognition system," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 1993, vol. II, pp. 463–466.
[37] J. Wilpon, L. Rabiner, C.-H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 38, no. 11, pp. 1870–1990.

Mazin G. Rahim (S'86–M'91) received the B.Eng. and Ph.D. degrees from the University of Liverpool, England, in 1987 and 1991, respectively.

He is currently a Principal Technical Staff Member at AT&T Research Laboratories, Murray Hill, NJ, where he is pursuing research in the areas of robustness and utterance verification for automatic speech recognition. Prior to joining AT&T, he was a Research Professor with the Center for Computer Aids for Industrial Productivity at Rutgers University, New Brunswick, NJ, where he was engaged in research in neural networks for speech and speaker recognition. He has numerous publications in the area of speech processing and is the author of *Artificial Neural Networks for Speech Analysis/Synthesis* (London: Chapman and Hall, 1994).

Dr. Rahim is currently an associate editor for the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. He is an Associate Member of the British Institute of Electrical Engineers (IEE).

Chin-Hui Lee (S'79–M'81–SM'90), for photograph and biography, see this issue, p. 265.

Biing-Hwang Juang (S'79–M'81–SM'87–F'92), for photograph and biography, see this issue, p. 265.