A Dynamic In-Search Data Selection Method With Its Applications to Acoustic Modeling and Utterance Verification

Hui Jiang, *Member, IEEE*, Frank K. Soong, *Member, IEEE*, and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—In this paper, we propose a dynamic in-search data selection method to diagnose competing information automatically from speech data. In our method, the Viterbi beam search is used to decode all training data. During decoding, all partial paths within the beam are examined to identify the so-called competing-token and true-token sets for each individual hidden Markov model (HMM). In this work, the collected data tokens are used for acoustic modeling and utterance verification as two specific examples. In acoustic modeling, the true-token sets are used to adapt HMMs with a sequential maximum *a posteriori* adaptation method, while a generalized probabilistic descent-based discriminative training method is proposed to improve HMMs based on competing-token sets. In utterance verification, under the framework of likelihood ratio testing, the true-token sets are employed to train *positive* models for the *null* hypothesis and the competing-token sets are used to estimate *negative* models for the *alternative* hypothesis. All the proposed methods are evaluated in Bell Laboratories communicator system. Experimental results show that the new acoustic modeling method can consistently improve recognition performance over our best maximum likelihood estimation models, roughly 1% absolute reduction in word error rate. The results also show the new verification models can significantly improve the performance of utterance verification over the conventional anti models, almost relatively 30% reduction of equal error rate when identifying misrecognized words from the recognition results.**

***Index Terms*—Competing token, discriminative training, in-search data selection, log likelihood ratio (LLR) testing, sequential maximum a posteriori (MAP) adaptation, true token (TT).**

I. Introduction

IN THE past decade, automatic speech recognition (ASR) has been significantly improved across almost all different tasks, from digit recognition to very large vocabulary broadcast news transcription. These impressive progresses can be attributed to many factors. Among many others, one is that the powerful statistical model, namely the hidden Markov model (HMM), has been broadly adopted as a fundamental tool to represent speech signals, which can be automatically learned from training data. Another important reason is that more and more speech corpus is becoming available for public use in the speech community, which assures a reliable estimation of large-scale HMM sets. At present, data collection has been commonly regarded as an initial and indispensable step to build a successful ASR system. For a state-of-the-art large vocabulary ASR system, it is not rare that the system needs to be trained on hundreds of hours, or even more, of speech data. On the other hand, training procedure of HMM has already matured as a standard routine, which is a parameter estimation based on maximum likelihood (ML) criterion with a flexible model parameter tying [23], [24]. In the standard training procedure, all speech data collected under various conditions are usually pooled together to estimate HMM parameters for each speech unit. As we are able to access more and more training data, limitations of the standard "pool-and-estimate" strategy become apparent. In many situations, people have noticed that the performance of an ASR system usually is saturated after the amount of total training data exceeds a certain point. Therefore, in order to make a better use of the huge amount of data available today, it is strongly desirable to have a more intelligent method to analyze tons of data and explore additional pertinent information for our recognition and other modeling purposes.

Beyond the simple "pool-and-estimate" training strategy, several researchers have recently proposed other training approaches which are usually based on a preliminary analysis of training data. In [2], the so-called "selective training" has been proposed to weight data tokens differently in training procedure according to an log likelihood ratio (LLR) based confidence measure. In this way, the influence of some outliers can probably be eliminated and more robust model estimation can be achieved. Moreover, in [1], a normalization method is proposed to compensate speaker variations in training procedure to get the so-called "compact" model. Also, in the cluster adaptive training (CAT) method proposed in [6], all training data is first clustered into several classes, then the same strategy is employed to normalize inter-cluster variations to achieve some cluster-dependent canonical models. Along this line, some further extensions have already been reported, such as normalizing "irrelevant" variability to learn model structure (state tying) from data in [10], a Bayesian approach to combine canonical models of CAT in [11], [13], etc.

In this paper, we propose a data analysis procedure to dynamically diagnose competing information available in speech data. In our approach, the Viterbi search process is used to decode every training utterance just as in speech recognition to automatically collect competing information from data. In other words, for

Manuscript received December 21, 2001; revised May 26, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ramesh A. Gopinath.

H. Jiang is currently with Department of Computer Science, York University, Toronto, ON M3J 1P3 Canada (e-mail: hj@cs.yorku.ca).

K. Soong is with Microsoft Research Asia (MSRA), Beijing, China (e-mail: frankkps@microsoft.com).

C. H. Lee is with School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail:chin.lee@ece.gatech.edu).

Digital Object Identifier 10.1109/TSA.2005.851947

each HMM, two different data sets, namely *competing* token set and *true* token set, are automatically collected during the search procedure. Then, the competing information collected from the in-search process can be used for many different purposes. In this paper, as two specific examples, the competing information is applied to acoustic modeling and utterance verification. In the first application of acoustic modeling [14], data tokens in *true* token sets are used to enhance original models by sequential Bayesian adaptation while data tokens in *competing* token sets are used to improve model discrimination capability by a generalized probabilistic descent (GPD) iterative algorithm. In the GPD-based discriminative training, we formulate objective function as the total number of the so-called *impostor* words[1] appearing during the Viterbi decoding. In this way, by explicitly incorporating the competing information gathered in the proposed in-search data analysis procedure, we are able to enhance discrimination capability of original acoustic models and, in turn, improve speech recognition performance. In the second application of utterance verification [12], the collected competing information is used to improve utterance verification in large-vocabulary continuous speech recognition. Concretely, we use all tokens in the *true* token set to estimate the so-called *positive* model and ones in the *competing* token set for its corresponding *negative* model. Then, we perform utterance verification in the framework of LLR testing, i.e., the confidence measure of each recognized segment is calculated as the LLR between its *positive* and *negative* models. Furthermore, the confidence scores for small segments, such as phonemes, can be combined to obtain confidence measures for the whole word and/or utterance in order to make verification decision in word or utterance level. In this paper, the proposed methods are evaluated in the Bell Laboratories' DARPA Communicator system. In the first part of experiments on acoustic modeling, the proposed method is used to improve our ML-trained acoustic models. The experimental results show that our new acoustic model training approach which considers the competing information collected in the in-search data selection method can significantly improve speech recognition performance over our best ML-trained HMM models. An absolute 1% improvement in word error rate (WER) over the best ML models are consistently observed in two different testing sets. In the second part of experiments on utterance verification, when compared with the standard anti models, which are trained from the forced-alignment phone segmentation, the positive and negative model which are trained on the corresponding data tokens sets collected in the in-search data selection method yields a much better verification performance in terms of identifying misrecognized words from the output of our baseline recognizer. When verifying correctly recognized words versus misrecognized words in the baseline recognition system, we have achieved around 30% relative reduction of equal error rate (EER) with the new verification models.

The remainder of the paper is organized as follows. In Section II, at first, we give the definition of competing tokens with respect to a set of predefined models (or classes). Next, in Section III, we introduce the in-search data selection method which can automatically collect competing tokens from speech data. As the first application example, we show how we can enhance acoustic modeling with the collected competing information in Section IV. Then, in Section V, as another example, the collected competing information is used for utterance verification in large-vocabulary continuous speech recognition. In Section VI, the proposed methods are evaluated in Bell Laboratories communicator system and all experimental results are reported. Finally, we conclude the paper with our findings and discussions in Section VII.

II. Competing Versus True Token

Given a set of classes (or models), in this section, we first define the competing tokens with respect to these models. For any typical pattern classifier, given an observation X as input, it always gives a class W as its output. However, X could come from several different sources: 1) X actually comes from the class W, i.e., a correct classification; 2) X comes from other competing classes of W, i.e., a classification error; or 3) X is an outlier, i.e., X comes from none of classes registered in the classifier. Here, if an observation X is classified as W but it actually does not belong to the class W, we simply call it as a *competing token* (CT) of the class W. On ther other hand, if X actually belongs to W, it is called a *true token* (TT) of the class W.

In speech recognition, we encounter the same situation. Given a speech utterance X as input, a speech recognizer usually gives a linguistic unit W as output.[2] However, the input X itself could be a TT of class W or just a CT of word W. A classical speech recognizer does not explicitly provide information to differentiate whether X is a TT or CT of W. If the speech recognizer is based on the optimal Bayes decision rule, we can define the set of all competing tokens of W as $\mathcal{S}_C(W)$

$$\mathcal{S}_C(W) = \{\mathcal{Y} | \Pr(W|\mathcal{Y}) > \Pr(W'|\mathcal{Y}),\ \forall\, W' \neq W \\ \mathcal{Y} \not\sqsupseteq W,\ \text{and}\ \Pr(W|\mathcal{Y}) \geq \xi\} \quad (1)$$

where $\mathcal{Y} \not\sqsupseteq W$ denotes that $\mathcal{Y}$ is not from class W and $\xi > 0$ is a constant. The set $\mathcal{S}_C(W)$ is called the CT set of word W. In the above definition, any competing token with too small observation probability $(< \xi)$ are excluded from our definition because they are usually negligible in the following discussions.

On the other hand, the set of all true tokens of W is denoted as $\mathcal{S}_T(W)$

$$\mathcal{S}_T(W) = \{\mathcal{X} | \Pr(W|\mathcal{X}) > \Pr(W'|\mathcal{X}) \\ \forall\, W' \neq W\ \text{and}\ \mathcal{X} \sqsupseteq W\} \quad (2)$$

where $\mathcal{X} \sqsupseteq W$ stands for that the token $\mathcal{X}$ belongs to class W. For convenience, the set $\mathcal{S}_T(W)$ is called the TT set of word W.

Given an observation $\mathcal{X}$ from either $\mathcal{S}_C(W)$ or $\mathcal{S}_T(W)$ as input, the speech recognizer will equally give W as its recognized result. The Bayes decision procedure in the recognizer usually is unable to provide enough information to differentiate whether $\mathcal{X}$ belongs to $\mathcal{S}_C(W)$ or $\mathcal{S}_T(W)$.

III. Dynamic Data Selection in Search

In isolated speech recognition, given a recognizer and a training database, it is straightforward to define $\mathcal{S}_C(W)$ and $\mathcal{S}_T(W)$ for every isolated word W. We recognize each isolated

[1] The definition of *impostor* words will be given in Section IV-A.

[2] W may be a phone, a syllable, a word, a phrase, or a sentence.

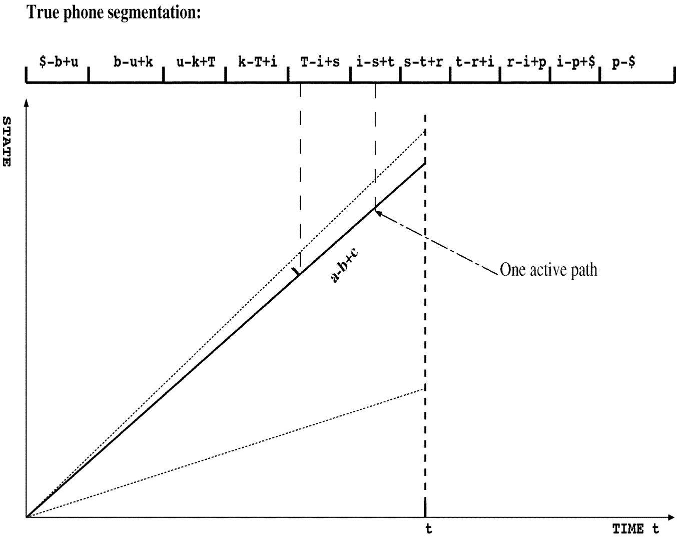

Fig. 1. Data selection procedure during Viterbi beam search in continuous speech recognition is illustrated.

speech token X in training data with the recognizer. Assume the recognition result is W, if the true label of X actually is W, then this token is assigned to the TT set $\mathcal{S}_T(W)$ of W; otherwise, it is assigned to the CT set $\mathcal{S}_C(W)$ of W. However, in continuous speech recognition, it becomes much more difficult to define the *CT* or *TT* set because of unknown unit boundaries. For example, in large-vocabulary, continuous speech recognition, it is very hard to associate a definite part of speech data to the CT set because numerous boundaries are possible and they are all considered during the Viterbi decoding. As a result, any possible segmentation in an utterance could potentially become a competing token. Obviously, an exhaustive search is too expensive to be affordable. In this paper, we propose an efficient way to identify *CT* and *TT* for different speech units in continuous speech recognition. In our method, every utterance in training database is recognized with Viterbi beam search algorithm just as in regular recognition phase. During Viterbi search, all potential segments located in all active partial paths within the search beam width are examined to identify *CT* and *TT* sets for every HMM model. We know, all partial paths surviving during beam search always have relatively large likelihood values and usually potentially compete with the true path. The basic idea here is that we only examine the subset of all active partial paths and view them as potential candidates for CTs or TTs. In this way, the decision procedure in speech recognition is simulated and a much richer competing information embedded in the data will be investigated.

Given a speech recognizer and a training database, for every utterance $\mathbf{X}$ in database, we first generate a reference segmentation by forced-aligning the utterance with its reference transcription. Next, we perform Viterbi beam search to recognize the utterance $\mathbf{X}$. During the search, at every time instant t, we backtrack every word-ending active partial path and compare all subword segments (usually phone) with the reference segmentation to determine whether each particular segment should be assigned to TT set $\mathcal{S}_T$ or CT set $\mathcal{S}_C$. The above procedure is carried out for all training data to collect two token sets, e.g., $\mathcal{S}_T(a)$ and $\mathcal{S}_C(a)$, for every subword HMM model a. Without losing generalization, hereafter, we only discuss the case where tri-phone HMMs are used for recognition. We denote the ith active word-ending path at the time instant t as $\mathcal{L}_i(t)$. We only back trace the most recently decoded word W in the partial path $\mathcal{L}_i(t)$ to get all its tri-phone segments. Assume the word W in $\mathcal{L}_i(t)$ consists of M different tri-phone segments as

$$\mathcal{Q}_{t_1}^{t_{M+1}}(W) = \left\{\mathcal{P}_{t_1}^{t_2}(a_1)\mathcal{P}_{t_2}^{t_3}(a_2)\cdots\mathcal{P}_{t_M}^{t_{M+1}}(a_M)\right\} \qquad (3)$$

where $\mathcal{Q}_{t_1}^{t_{M+1}}(W)$ represents the whole segment corresponds to the word W, starting at time t_1 and ending at t_{M+1}, and $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)(1 \leq m \leq M)$ stands for the mth tri-phone segment with tri-phone id a_m, starting time t_m, and ending time t_{m+1}. Then, for every tri-phone segment $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)(1 \leq m \leq M)$, we compare it with the reference phone segmentation generated from the forced-alignment procedure. If the tri-phone segment $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)$ matches well with any in the reference segmentation, then we view it as a true token of the tri-phone a_m and is assigned to the TT set $\mathcal{S}_T(a_m)$ accordingly. Otherwise, it is thought as the competing token of tri-phone a_m and assigned to the CT set $\mathcal{S}_C(a_m)$. The whole data selection procedure is illustrated in Fig. 1, where, at time instant t, each hypothesized token, i.e., triphone segment $\mathrm{a} - \mathrm{b} + \mathrm{c}$, is compared with all tokens in the reference segmentation to determine which set, $\mathcal{S}_T(\mathrm{a} - \mathrm{b} + \mathrm{c})$ or $\mathcal{S}_C(\mathrm{a} - \mathrm{b} + \mathrm{c})$, it should be assigned to.

In this paper, the matching procedure between two phone segments is implemented based on the overlap between them. Given a hypothesized tri-phone segment, we calculate its maximum overlap rate over all reference segments with the same tri-phone identity in the reference segmentation. For instance, if the hypothesized segment $\mathcal{P}_{t_s}^{t_e}(a_m)$ has the same tri-phone identity a_m as a reference segment $\bar{\mathcal{P}}_{t'_s}^{t'_e}(a_m)$, then the overlap rate between them is calculated as

$$\Psi = \frac{\min\{t_e, t'_e\} - \max\{t_s, t'_s\} + 1}{\frac{((t_e - t_s + 1) + (t'_e - t'_s + 1))}{2}}. \qquad (4)$$

Obviously, the range of Ψ is $(-\infty, 1]$. If two segments $\mathcal{P}$ and $\bar{\mathcal{P}}$ intersect, then $\Psi \geq 0$; otherwise, $\Psi < 0$. If $\mathcal{P}$ and $\bar{\mathcal{P}}$ are identical, then $\Psi = 1$. Then the maximum overlap rate of a hypothesized segment is computed over all reference segment with the same tri-phone identity a_m in the reference segmentation. If the maximum overlap rate of the hypothesized phone segment $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)$ exceeds a threshold ξ_1, then we decide $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)$ matches with the reference label and $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)$ is assigned to the *TT* set $\mathcal{S}_T(a_m)$ of triphone a_m. Otherwise, if the maximum overlap rate of the hypothesized phone segment $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)$ is below a threshold ξ_2 and $l_{t_m}^{t_{m+1}} > h_{t_m}^{t_{m+1}}$, where $l_{t_m}^{t_{m+1}}$ denotes the average likelihood per frame of the hypothesis $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)$ and $h_{t_m}^{t_{m+1}}$ is the average likelihood per frame of the same segment based on the forced alignment procedure, then we consider the hypothesis $\mathcal{P}_{t_m}^{t_{m+1}}(a_m)$ as a competing token of the triphone and it is assigned to the *CT* set $\mathcal{S}_C(a_m)$.

After we go through all utterances in training set, we will have two token sets for each tri-phone model a:[3] $\mathcal{S}_C(a)$ and $\mathcal{S}_T(a)$, where $\mathcal{S}_C(a)$ consists of all *competing* tokens of triphone model a and $\mathcal{S}_T(a)$ contains all its *true* tokens. Obviously, these two different data sets can be used for many different purposes. In this paper, two applications will be discussed below. First, in Section IV, we will show how they can be used to improve recognition acoustic HMM models. Second, in Section V, these two data sets will be used to estimate verification models to improve the performance of utterance verification in large vocabulary speech recognition.

IV. Application I: Acoustic Modeling

At first, we assume each speech unit, to say a tri-phone, is modeled by an N-state CDHMM with parameter vector $\Lambda = (\pi, A, \theta)$, where π is the initial state distribution, $A = \{a_{ij} | 1 \leq i, j \leq N\}$ is transition matrix, and θ is parameter vector composed of mixture parameters $\theta_i = \{\omega_{ik}, m_{ik}, r_{ik}\}_{k=1,2,\cdots,K}$ for each state i, where K denotes number of Gaussian mixtures in each state. The state observation p.d.f. is assumed to be a mixture of multivariate Gaussian distribution with diagonal precision matrix

$$\begin{aligned} p(\mathbf{x}|\theta_i) &= \sum_{k=1}^{K} \omega_{ik} \mathcal{N}(\mathbf{x}|m_{ik}, r_{ik}) \\ &= \sum_{k=1}^{K} \omega_{ik} \prod_{d=1}^{D} \sqrt{\frac{r_{ikd}}{2\pi}} e^{-\frac{1}{2} r_{ikd}(x_d - m_{ikd})^2} \end{aligned} \tag{5}$$

where mixture weights ω_{ik}s satisfy the constraint $\sum_{k=1}^{K} \omega_{ik} = 1$.

Once we have collected two sets of data, namely $\mathcal{S}_C(a)$ and $\mathcal{S}_T(a)$, for each tri-phone model Λ_a, it is possible to adjust original acoustic models to improve their discrimination capability to enhance speech recognition performance. Intuitively, for every *true* token in $\mathcal{S}_T$, its corresponding tri-phone model is adapted to increase the likelihood of the observed data given this model so that the model can be pushed toward correct recognition results. Concretely, the *maximum a posteriori* (MAP) based sequential Bayesian learning method can be used to adapt the model parameters based on data tokens in *TT* set $\mathcal{S}_T$. On the other hand, for every *competing* token in $\mathcal{S}_C$, the model is tuned to decrease the likelihood of the observation data so that the competing token could be dragged out of beam width in Viterbi search. Concretely, the GPD based discriminative training is employed to minimize the total number of the so-called *impostor* words appearing during decoding. In the following, model parameters adjusting algorithm based on the GPD discriminative training will be derived. For simplicity, in this paper, only mean and variance vectors of CDHMM are updated and other parameters remain constant. As for the MAP-based sequential Bayesian learning with TT sets, refer to [9] for details.

A. GPD-Based Discriminative Training With CT Sets

In order to improve discrimination capability of HMM models, for every token in the *CT* set, intuitively, we should adjust model parameters to decrease its likelihood value given the model. Among all wrong words which appears during Viterbi decoding, if the likelihood value of the word segment significantly exceeds its likelihood value given its reference model, it will become much more likely for it to finally emerge in recognition results to actually become a misrecognized word. Here, we call this word an *impostor* word. As one possibility, we can choose objective function of the discriminative training as the total number of *impostor* words appearing during Viterbi decoding. If we can minimize the total number of *impostor* words, hopefully, we can reduce the final word error rate (WER) of recognition in an indirect way.

Suppose we have a wrong word W appearing during decoding, by backtracking the partial path we get the segmentation information of this word W as $\mathcal{Q}_{t_1}^{t_M}$, which starts from time t_1 and ends at time t_M. Then the misclassification distance measure for the whole word segment $\mathcal{Q}_{t_1}^{t_M}$ is defined as

$$d_W = -l\left(\mathcal{Q}_{t_1}^{t_M} | \Lambda_{ref}(\mathcal{Q})\right) + l\left(\mathcal{Q}_{t_1}^{t_M} | \Lambda_W\right) \tag{6}$$

where $l(\cdot)$ denotes log likelihood function, and $\Lambda_{ref}(\mathcal{Q})$ stands for the reference model for the segment $\mathcal{Q}_{t_1}^{t_M}$ according to the optimal Viterbi path obtained in forced-alignment against reference transcription, and Λ_W is the connected HMMs for the word W. Here, we define W as an impostor word if the above misclassification measure $d_W > 0$. Next, the misclassification measure d_W is embedded in a sigmoid function to approximate the zero-one decision of counting the impostor word. A general form of the "smoothed" count of the *impostor* word is defined as

$$\ell(d_W) = \frac{1}{1 + \exp(-\gamma \cdot d_W + \vartheta)} \tag{7}$$

where ϑ and γ are set to control the shape of sigmoid function. Then, we can sum up the above "smoothed" count over all wrong words, which appear during the Viterbi search, to calculate the total number of *impostor* words as

$$L(\vec{\Lambda}) \propto \sum_{W} \ell(d_W) \tag{8}$$

where $\vec{\Lambda}$ denotes all parameters in the given HMM set. Then, the function $L(\vec{\Lambda})$ is treated as an objective function in our discriminative training. All HMM parameters $\vec{\Lambda}$ in the given HMM set are estimated based on the minimization of the above "smoothed" count of total *impostor* words when we decode all

[3] A mechanism is implemented to guarantee that the same segment $\mathcal{P}_t^{t'}(a)$ will never occur in the same set more than once.

utterances in training data set. Here, the so-called GPD algorithm is adopted to minimize the above objective function. In a GPD-based minimization algorithm, the target function L is minimized according to an iterative procedure

$$\Lambda_{t+1} = \Lambda_t - \epsilon_t \cdot \nabla L(\vec{\Lambda})|_{\Lambda=\Lambda_t} \tag{9}$$

where ϵ_t is step size, and $\nabla L(\vec{\Lambda})|_{\Lambda=\Lambda_t}$ is the gradient function of the target function at $\Lambda = \Lambda_t$.

When we update model parameters based on the above GPD algorithm, we only adjust the HMM parameters related to competing tokens which have been collected in the CT sets. Given any wrong word W, assume the segmentation $\mathcal{Q}(W)$ of W consists of a sequence phoneme tokens, i.e., $\mathcal{Q}(W) = \{\mathcal{Y}_1(a_1)\ \mathcal{Y}_2(a_2)\ \cdots \mathcal{Y}_M(a_M)\}$. Among all of these tokens, we will only update HMM parameters related to competing tokens. For example, assume any one competing token, to say $\mathcal{Y}_m(a_m)$, of tri-phone a_m. We will use the above GPD to modify the tri-phone HMM model Λ_{a_m} and its reference model, denoted as $\bar{\Lambda}_{\mathcal{Y}}$, corresponding to $\mathcal{Y}_m(a_m)$. Please note that $\bar{\Lambda}_{\mathcal{Y}}$ denotes the reference HMM for only one phone segment $\mathcal{Y}_m(a_m)$ while $\Lambda_{ref}(\mathcal{Q})$ in (6) represents the reference HMMs for a whole word segment. Assume the tri-phone model Λ_{a_m} is an N-state CDHMM with parameter vector $\Lambda = (\pi, A, \theta)$ and the reference model $\bar{\Lambda}_{\mathcal{Y}}$ is an M-state[4] CDHMM with parameter vector $\bar{\Lambda} = (\bar{\pi}, \bar{A}, \bar{\theta})$, where $\bar{\theta}$ is composed of mixture parameters $\bar{\theta}_i = \{\bar{\omega}_{ik}, \bar{m}_{ik}, \bar{r}_{ik}\}_{k=1,2,\cdots,K}$ for each state i, and state observation density $p(\mathbf{x}|\bar{\theta}_i)$ has the same form as (5). Following the same procedure in [16] and [17], we can derive all formula to update HMM parameters based on (9). As we have mentioned, among all CDHMM parameters, only mean and precision vectors are updated in this paper. Moreover, the above GPD algorithm is flexible enough to run in either batch or incremental mode. In batch mode, the gradient descent of each HMM parameter is first accumulated over all collected competing tokens, and then the parameter is modified in a following stage. In incremental mode, the gradient descent of HMM parameters is calculated for every competing token, and the HMM parameters are updated immediately on a token-by-token basis. The GPD-based HMM parameter updating formula for both batch and incremental modes will be given in Sections IV-B and IV-C, respectively.

B. Batch Updating Algorithm

In batch mode, given any data set, we first run the in-search data selection method, described in Section III, to prepare two data token sets $\mathcal{S}_T(a)$ and $\mathcal{S}_C(a)$ for each tri-phone model Λ_a beforehand. During the token selection procedure, for each competing token $\mathcal{Y}$, we also attach it with a misclassification measure $d_W(\mathcal{Y})$ of the word W to which the token $\mathcal{Y}$ belongs. Then, the whole triphone HMM model set can be updated based on all collected tokens in a separate round. For example, given a competing token $\mathcal{Y}(a) = \{y_1, y_2, \cdots, y_T\}$ in the CT set $\mathcal{S}_C(a)$ of tri-phone a, we assume $\{s_1, s_2, \cdots, s_T\}$ is its corresponding optimal Viterbi path in triphone model Λ_a, and $\{l_1, l_2, \cdots, l_T\}$ is its optimal mixture component label sequence. Then, the mean and variance vectors $\{m_{ik}, r_{ik} | 1 \leq i \leq N, 1 \leq k \leq K\}$ of tri-phone model Λ_a are updated as follows:

$$\begin{aligned} m''_{ik} &= m'_{ik} - \epsilon_1 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \left. \frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial m_{ik}} \right|_{m_{ik}=m'_{ik}} \\ &= m'_{ik} - \epsilon_1 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \gamma \ell(d_W)\left(1 - \ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T} \left(y_t - m'_{ik}\right) \delta(s_t - i) \delta(l_t - k) \end{aligned} \tag{10}$$

$$\begin{aligned} \log r''_{ik} &= \log r'_{ik} - \epsilon_1 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \left. \frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial \log r_{ik}} \right|_{r_{ik}=r'_{ik}} \\ &= \log r'_{ik} - \epsilon_1 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \gamma \ell(d_W)\left(1 - \ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T} \left[1 - r'_{ik}\left(y_t - m'_{ik}\right)^2\right] \\ &\quad \times \delta(s_t - i)\delta(l_t - k) \end{aligned} \tag{11}$$

where $\delta(\cdot)$ denotes the Kronecher delta function.

As for the reference model $\bar{\Lambda}_{\mathcal{Y}}$, we assume that the optimal state path and Gaussian mixture component sequence are denoted as $\{\bar{s}_1, \bar{s}_2, \cdots, \bar{s}_T\}$ and $\{\bar{l}_1, \bar{l}_2, \cdots, \bar{l}_T\}$ for the token $\mathcal{Y}(a)$. Similarly, its mean and variance vectors are updated as

$$\begin{aligned} \bar{m}''_{ik} &= \bar{m}'_{ik} - \epsilon_2 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \left. \frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial \bar{m}_{ik}} \right|_{\bar{m}_{ik}=\bar{m}'_{ikd}} \\ &= \bar{m}'_{ik} + \epsilon_2 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \gamma \ell(d_W)\left(1 - \ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T} \left(y_t - \bar{m}'_{ik}\right) \delta(\bar{s}_t - i) \delta(\bar{l}_t - k) \end{aligned} \tag{12}$$

$$\begin{aligned} \log \bar{r}''_{ik} &= \log \bar{r}'_{ik} - \epsilon_2 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \left. \frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial \log \bar{r}_{ik}} \right|_{\bar{r}_{ik}=\bar{r}'_{ik}} \\ &= \log \bar{r}'_{ik} + \epsilon_2 \sum_{\mathcal{Y} \in \mathcal{S}_C(a)} \gamma \ell(d_W)\left(1 - \ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T} \left[1 - \bar{r}'_{ik}\left(y_{td} - m'_{ik}\right)^2\right] \\ &\quad \times \delta(\bar{s}_t - i)\delta(\bar{l}_t - k). \end{aligned} \tag{13}$$

Above, ϵ_1 and ϵ_2 are two different step sizes for competing model and reference model, which are set up manually in experiments. As we see, in the batch mode, HMM parameters are modified only after we have considered all competing tokens. Please note that during the GPD-based model updating the word-level misclassification measure d_W is used for each competing token. This leads to minimization of total *impostor* words in Viterbi decoding. Finally, the whole batch algorithm, including GPD training and MAP adaptation, is shown in Algorithm 1.

[4] M maybe varies for different tokens according to the forced-alignment result.

```
Algorithm 1 Batch Updating Algorithm
repeat
  1) Token Collection: The data selection algorithm in Section III is performed
     to collect competing token set $S_C(a)$ and true token set $S_T(a)$ for every
     triphone HMM $a$ based on the current model set $\vec{\Lambda}$.
  2) GPD-based Discriminative Training:
  for each triphone model $\Lambda_a$ in HMM set do
    Update $\Lambda_a$ based on the CT set $S_C(a)$ according to batch GPD algorithm shown in (10)-(13).
  end for
  3) Sequential MAP Adaptation:
  for each triphone model $\Lambda_a$ in HMM set do
    for each token in set $S_T(a)$ do
      Adapt triphone model $\Lambda_a$ with sequential MAP adaptation method.
    end for
  end for
until some conditions are met.
```

The model updating algorithm can be run on original training data to refine the HMM set which is estimated by normal maximum likelihood estimation (MLE) training procedure or on a new set of adaptation data to adjust the original HMM set into a new condition. If this algorithm is used to adapt an existing HMM set to a new condition based on some adaptation data, it becomes one of the so-called *discriminative adaptation* approaches [7]. Obviously, the batch updating procedure can be repeated iteratively. In the next iteration, we usually have to recollect token sets based on the model set modified in the last iteration.

C. Incremental Updating Algorithm: Embedded in Search

Instead of collecting all competing tokens beforehand in a batch mode, we can also embed the whole model updating algorithm into Viterbi search procedure and adjust model parameters sequentially on a token-by-token basis. Given an utterance in the training or adaptation set, we perform Viterbi beam search based on the current HMM models. During the search, the data selection method described in Section III is performed to choose competing tokens in all active word-ending paths. Once a competing token, to say the nth token, $\mathcal{Y}_n(a) = \{y_1, y_2, \cdots, y_T\}$, is identified, its corresponding tri-phone model Λ_a and reference model $\bar{\Lambda}_{\mathcal{Y}}$ are updated as follows:

$$\begin{aligned} m_{ik}^{(n+1)} &= m_{ik}^{(n)} - \epsilon_1 \cdot \left.\frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial m_{ik}}\right|_{m_{ik}=m_{ik}^{(n)}} \\ &= m_{ik}^{(n)} - \epsilon_1 \cdot \gamma\ell(d_W)\left(1-\ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T}\left(y_t - m_{ik}^{(n)}\right)\delta(s_t - i)\delta(l_t - k) \end{aligned} \tag{14}$$

$$\begin{aligned} \log r_{ik}^{(n+1)} &= \log r_{ik}^{(n)} - \epsilon_1 \cdot \left.\frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial \log r_{ik}}\right|_{r_{ik}=r_{ik}^{(n)}} \\ &= \log r_{ik}^{(n)} - \epsilon_1 \cdot \gamma\ell(d_W)\left(1-\ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T}\left[1 - r_{ik}^{(n)}\left(y_t - m_{ik}^{(n)}\right)^2\right] \\ &\quad \times \delta(s_t - i)\delta(l_t - k) \end{aligned} \tag{15}$$

$$\begin{aligned} \bar{m}_{ik}^{(n+1)} &= \bar{m}_{ik}^{(n)} - \epsilon_2 \cdot \left.\frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial \bar{m}_{ik}}\right|_{\bar{m}_{ik}=\bar{m}_{ik}^{(n)}} \\ &= \bar{m}_{ik}^{(n)} + \epsilon_2 \cdot \gamma\ell(d_W)\left(1-\ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T}\left(y_t - \bar{m}_{ik}^{(n)}\right)\delta(\bar{s}_t - i)\delta(\bar{l}_t - k) \end{aligned} \tag{16}$$

$$\begin{aligned} \log \bar{r}_{ik}^{(n+1)} &= \log \bar{r}_{ik}^{(n)} - \epsilon_2 \cdot \left.\frac{\partial \ell\left(d_W(\mathcal{Y})\right)}{\partial \log \bar{r}_{ik}}\right|_{\bar{r}_{ik}=\bar{r}_{ik}^{(n)}} \\ &= \log \bar{r}_{ik}^{(n)} + \epsilon_2 \cdot \gamma\ell(d_W)\left(1-\ell(d_W)\right) \\ &\quad \times \sum_{t=1}^{T}\left[1 - \bar{r}_{ik}^{(n)}\left(y_n - m_{ik}^{(n)}\right)^2\right] \\ &\quad \times \delta(\bar{s}_t - i)\delta(\bar{l}_t - k). \end{aligned} \tag{17}$$

Please note that in the above equations GPD training is still based on the word-level misclassification measure d_W. On the other hand, if a true token $\mathcal{X}$ is identified, the sequential Bayesian method is used to update the corresponding HMM. The whole on-line updating algorithm is shown in Algorithm 2.

```
Algorithm 2 Incremental Updating Algorithm
Repeat
  for each utterance in training set do
    Forced alignment against transcription to get reference phone segmentation
    Perform Viterbi beam search based on current HMM set
    for every time instant do
      Backtrack all active word-ending partial paths
      for each active word-ending partial path do
        —Calculate misclassification measure $d_W$ for the most recently decoded word segment $\mathcal{Q}(W)$
        —Back trace all phoneme segments in $\mathcal{Q}(W)$.
        for each phoneme segment $\mathcal{Y}(a)$ in $\mathcal{Q}(W)$ do
          if it is a competing token then
            Update the tri-phone model $\Lambda_a$ and the reference model $\bar{\Lambda}_{\mathcal{Y}}$ according to (14)-(17).
          else if it is a true token then
            Updating tri-phone model $\Lambda_a$ by the sequential MAP estimation.
          end if
        end for
      end for
    end for
  end for
until Some converge conditions are met.
```

As a remark, discriminative training has been studied for years in speech community, mainly for small or medium vocabulary tasks. Recently, MMIE-based discriminative training is applied to large vocabulary continuous speech recognition as in [7] and [26]. The work in this paper is another recent effort to apply discriminative training to large vocabulary continuous speech recognition under the framework of MCE/GPD.

V. Application II: Utterance Verification

In many practical applications, it becomes more desirable and urgent to equip a speech recognizer with a capability of *utterance verification* (UV) [19]. Utterance verification is a procedure used to verify how reliable the recognition results are. Usually, a quantitative score, also called *confidence measure*, is used to indicate the reliability of every recognition decision. Based on the confidence measure, a series of further actions can be taken after recognition, e.g., to reject or remedy the recognition results. Utterance verification is a crucial technique to make today's speech recognizers more "intelligent." For instance, a speech recognizer with a powerful UV capability will be able to smartly reject nonspeech noises, detect/reject out-of-vocabulary words, even correct some potential recognition mistakes, guide the system to perform unsupervized learning, and provide side information to assist high level speech understanding, etc.

Extensive studies on utterance verification have been performed recently in the literature. One of the most important progresses is to cast utterance verification scenario as a statistical hypothesis testing problem [19], [22]. According to the Neyman–Pearson lemma, an optimal test is to evaluate a likelihood ratio between two hypotheses, H_0 and H_1. However, the alternative hypothesis H_1 is a composite one and it consists of many heterogeneous events so that it is always very difficult to model H_1 appropriately in UV. In [19] and [22], the same HMM model structure is adopted to model H_1; they are commonly named *anti models*. Some limited successes have been obtained in using *anti models* to model the alternative hypothesis H_1 when *anti models* are trained from some discriminative training procedures. However, we are still in search of a more powerful method to model this complicated hypothesis.

In this section, we study the problem of utterance verification for large-vocabulary continuous speech recognition by using the collected token sets, $\mathcal{S}_C$ and $\mathcal{S}_T$. At first, we explain how to use the competing information to perform utterance verification in speech recognition.

A. Utterance Verification Based on Competing Information

Based on the explanation in Section II, given an observation X from either $\mathcal{S}_C(W)$ or $\mathcal{S}_T(W)$ as input, the speech recognizer will equally give W as its recognized result. The Bayes decision procedure in the recognizer usually is unable to present enough information on whether X belongs to $\mathcal{S}_C(W)$ or $\mathcal{S}_T(W)$. Obviously, the capability of utterance verification totally depends on how well we can distinguish $\mathcal{S}_C(W)$ from $\mathcal{S}_T(W)$. In this paper, statistical hypothesis testing is still adopted as a tool to separate $\mathcal{S}_C(W)$ from $\mathcal{S}_T(W)$ statistically. Given a recognition result W from a recognizer for the observation X, in order to reject or accept W, we test the *null* hypothesis

$$\mathbf{H_0} : X \in \mathcal{S}_T(W) \tag{18}$$

against the alternative hypothesis

$$\mathbf{H_1} : X \in \mathcal{S}_C(W). \tag{19}$$

Compared with the previous works on hypothesis testing in [19] and [22], both the *null* hypothesis H_0 and the alternative hypothesis H_1 in this work are well-defined from available data, which in turn will make our modeling problem easier. The simplest way to model $\mathcal{S}_T(W)$ and $\mathcal{S}_C(W)$ is to estimate two different models Λ_T and Λ_C for $\mathcal{S}_T(W)$ and $\mathcal{S}_C(W)$, respectively, based on all tokens collected from training data. Here, we call Λ_T the *positive* model and Λ_C the *negative* model. These models can be estimated from the collected token sets $\mathcal{S}_T(W)$ and $\mathcal{S}_C(W)$ according to different criteria, such as MLE or minimum verification error (MVE) [18], [20]. Once Λ_T and Λ_C are given, utterance verification is operated as the following likelihood ratio test

$$\eta = \frac{p(X|H_0)}{p(X|H_1)} = \frac{\Pr(X \in \mathcal{S}_T(W))}{\Pr(X \in \mathcal{S}_C(W))} = \frac{p(X|\Lambda_T)}{p(X|\Lambda_C)} \underset{H_1}{\overset{H_0}{\gtrless}} \tau \tag{20}$$

where τ is the decision threshold.

In this work, we choose subword HMM for both Λ_T and Λ_C, e.g., phones. In other words, for every phone, a positive and a negative models are estimated from its TT and CT sets which are collected in Section III. Given a speech recognition result, a confidence score based on likelihood ratio testing as in (20) is calculated for every phone segment by using the estimated positive and negative verification models. Then the scores of all phone segments within a word (or utterance) are averaged to get the confidence measure for the word (or utterance).

B. Training Positive and Negative Models

The first choice for Λ_T and Λ_C is mono-phone model. The tokens collected in $\mathcal{S}_T$ and $\mathcal{S}_C$ can be directly used to estimate positive and negative mono-phone model based on standard Baum–Welch algorithm. Furthermore, state-tying techniques in [23] and [24] can also be used to estimate state-tied tri-phone HMM model for Λ_T and Λ_C. The phoneme labels during data collection are used as phoneme identity for model training. For example, all tokens in the TT set $\mathcal{S}_T(a)$ are used for training positive model of tri-phone a, and all tokens in the CT set $\mathcal{S}_C(a)$ for negative model of tri-phone a.

Many other studies have also proposed to use the "competitors" information in recognition procedure to improve utterance verification, such as, N-Best in [21] and word-graph in [25]. None of these approaches uses any information in training data for verification. Our proposed method attempts to extract competing information from the available data. Another close work is the so-called *cohort* model [19], which is trained based on some predefined *corhort* sets, which are usually independent from the recognizer. As for the similarities and differences between the above method and other published algorithms, please refer to a recent survey paper in [15].

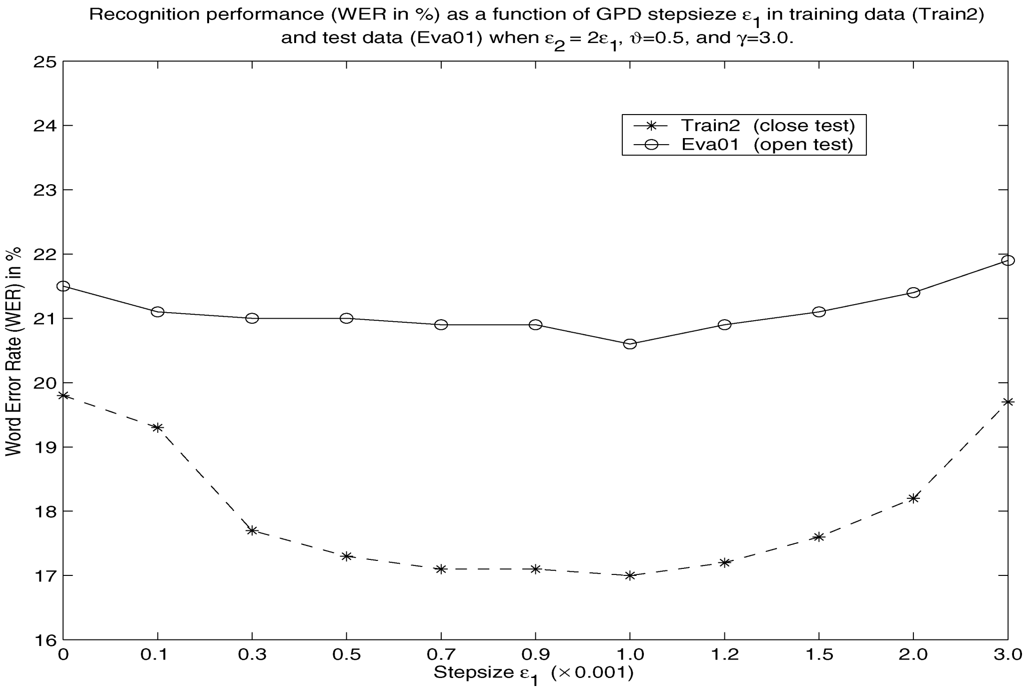

Fig. 2. Recognition performance (WER in %) as a function of GPD step size ϵ_1 in test sets *Train2* and *Eva01* when $\epsilon_2 = 2\epsilon_1$, $\vartheta = 0.5$, and $\gamma = 3.0$.

VI. Experiments

To examine the viability of the proposed methods, we evaluate them on the Bell Laboratories system of the DARPA Communicator task (travel reservation application). The proposed in-search data selection method is used to collect competing information from training data. In first part of experiments, the acoustic modeling methods in Section IV, both batch and incremental modes, are used to improve acoustic HMM models based on the collected token sets. In second part of experiments, the collected token sets are used to train verification models, both positive and negative models, in order to identify speech recognition errors in recognizer's outputs.

A. Experimental Setup

In our experiments, we use two different training sets. 1)*Train1*: It is a task-independent telephone database collected in Bell Laboratories and includes 34 h of telephone speech data. 2)*Train2*: It is a task-dependent (travel reservation) database collected during 2000-2001 DARPA evaluation and includes 12 h of speech data in total. We also use two different test sets for evaluation: 1) *Eva00*, which includes 1,395 utterances from data collection in year 2000, and 2) *Eva01*, which includes 4,000 utterances from data collection in year 2001. Both *Eva00* and *Eva01* are disjointed with the training sets.

In our recognition system, we used a 38-dimension feature vector, consisting of 12 Mel LPCCEP, 12 delta CEP, 12 delta-delta CEP, delta and delta–delta log energy. In the baseline system, acoustic HMM models are trained by using the standard Baum–Welch ML estimation on both *Train1* and *Train2* training sets, totally 46 h of speech data. The best ML-trained acoustic models are state-tied, tri-phone HMM models, which include roughly 4K distinct HMM states with an average of 13.2 Gaussian mixture components per state. In the following experiments on acoustic modeling, this best ML HMM model set is used as our initial model set. Besides, a class-based, tri-gram language model including 2,600 words is used in the baseline recognition system.

B. Experiments of Acoustic Modeling(I): Batch Mode

In the batch mode, the in-search data selection method described in Section III is first run to collect both CT and TT token sets from task-dependent training data set *Train2* based on the best ML HMM set. During token selection procedure, the decoder settings are different from those in baseline recognizer. Based on [26], in order to collect more representative tokens, we use a larger beam width, a weak language model (uni-gram), and a smaller LM weight. Under these settings, the data selection becomes quite slow, roughly ten times real-time (10 × RT), comparing to that the baseline system can run in real time during test phase. Some control parameters in data selection, e.g., ξ_1 and ξ_2, are set manually during experiments. The typical values for ξ_1 is 0.99 and for ξ_2 is between -20 and -30.

After data selection, we have collected two token sets, namely the competing token set $\mathcal{S}_C$ and the true token set $\mathcal{S}_T$, for each triphone HMM. Then the batch algorithm in Algorithm is run to update the initial acoustic HMM model set, which has been used for data selection. Please note the whole model updating procedure can be run very fast. The total running time is negligible comparing to the data selection procedure. This makes it much easier for us to manually optimize those control parameters related to GPD and sequential MAP, such as step size ϵ_1 and ϵ_2, ϑ, γ, and ε. As a reference, a typical setting for these control parameters are $\epsilon_1 = 0.001$, $\epsilon_2 = 0.002$, $\vartheta = 0.5$, $\gamma = 3.0$ and $\varepsilon = 30$. Among them, step size ϵ_1 and ϵ_2 are quite sensitive and should be set carefully. The other three ones are not very sensitive and can be roughly set in a certain range. In Fig. 2, we show recognition performance in training set *Train2* and test set

TABLE I
PERFORMANCE COMPARISON (WER IN PERCENTAGE) OF THE PROPOSED ACOUSTIC MODEL UPDATING ALGORITHM (BATCH MODE) WITH THE ORIGINAL ML MODEL IN TWO EVALUATION DATA SETS, NAMELY *Eva00* AND *Eva01*

iteration		*Eva00*	*Eva01*
0 (ML)		15.8	21.5
1	GPD	15.0	20.6
	GPD+MAP	**14.9**	**20.5**
2	GPD+MAP	14.9	20.7

TABLE II
PERFORMANCE COMPARISON (WER IN PERCENTAGE) OF THE ORIGNAL MLE MODEL, HMM MODELS OBTAINED FROM BATCH MODE OF MODEL UPDATING ALGORITHM, AND HMM MODELS FROM INCREMENTAL MODE (EMBEDDED IN SEARCH) OF MODEL UPDATING ALGORITHM IN THREE DIFFERENT EVALUATION DATA SETS: TRAIN1, EVA00, AND EVAL01

model	*Train2*	*Eva00*	*Eva01*
ML	19.8	15.8	21.5
Batch	16.9	14.9	20.5
Embed	16.7	15.0	20.7

TABLE III
EER COMPARISON (IN PERCENTAGE) IN *Train2* AND *Eva00* WHEN VERIFYING CORRECTLY RECOGNIZED WORDS AGAINST MISRECOGNIZED WORDS IN OUR BASELINE RECOGNIZER'S OUTPUT BASED ON DIFFERENT VERIFICATION MODELS

EER(%)	*Train2*	*Eva00*
std-anti-mono	37.9	40.0
new-mono	22.3	27.3
new-tri	17.1	24.6

Eva01 as a function of GPD step size ϵ_1. From the figure, we can see that our GPD-based discriminative training can effectively reduce word error rate in both training and testing data if a proper step size is chosen, though we minimize number of *impostor* words as our objective function in GPD training, and it is also shown that optimal step size is almost same for training set *Train2* and test set *Eva01*. Finally, the updated HMM set is compared with the original ML-trained in two evaluation data sets, *Eva00* and *Eva01*. The above HMM model updating algorithm can be run more than one iteration. In the next iteration, the data selection procedure must be repeated to generate new competing token sets for the new HMM model set. The experimental results are shown in Table I. We see that our best ML-trained HMM models achieve word error rate (WER) 15.8% and 21.5% for test sets *Eva00* and *Eva01*, respectively. In first iteration of model updating, if we only apply GPD-based discriminative training, denoted as *GPD* in Table I, we achieve WER 15.0% and 20.6% for *Eva00* and *Eva01*. If we apply both GPD training and MAP adaptation, denoted as $GPD+MAP$, we achieve WER 14.9% and 20.5% for *Eva00* and *Eva01*, respectively, which show almost 1% absolute improvement in WER. From these results, we can see that most part of improvement comes from GPD-based discriminative training. In Table I, we also list the results after two iterations of model updating. We do not obtain any improvement in the second iteration.

C. Experiments of Acoustic Modeling(II): Incremental Versus Bath Mode

In this section, we compare the batch model updating algorithm with the incremental model updating method. When we use the same setting for all control parameters in batch and incremental algorithm, the performance comparison is shown in Table II. In the table, as a reference, we also give the close-test results, i.e., the performance on the training data set *Train2*. Comparing with the batch mode, which achieves WER 16.9%, 14.9%, and 20.5% in test sets *Train1*, *Eva00*, and *Eva01*, respectively, the incremental method (embedded in search) gets a similar performance, i.e., 16.7% for *Train1*, 15.0% for *Eva00*, and 20.7% for *Eva01*. Both of them can achieve a significant improvement over the MLE models. Although batch and incremental methods achieve similar performance, they can be used in different occasions. In batch mode, the relatively time-consuming procedure, namely in-search data selection, is run only once and all collected tokens are saved. Thus, it is convenient to tune those control parameters related to GPD and MAP with the batch mode. Moreover, in batch mode, data selection procedure can run in parallel with many CPUs; thus, this makes it possible to process a relatively large database with the batch algorithm. On the other hand, once we have already known all control parameters, the incremental algorithm can update the whole HMM set in one pass without saving any intermediate results.

The results in Table II also raise another important issue. Both batch and incremental methods get a larger error reduction in training data, but the improvement in test set is much smaller. How to generalize the improvement to unseen test data becomes an important issue for future research.

D. Experiments of Utterance Verification

The UV method described in Section V is compared with the standard mono-phone anti models, which are trained with the fixed phone segmentation generated from forced alignment. In other words, all phone segments of a specific phone are collected to train the positive monophone verification model for this phone while all other phone segments are used to training negative (anti model) monophone model for this phone. The conventional verification models trained in this way are denoted as *std-anti-mono*. As for new verification models, we can creat mono-phone verification models (both positive and negative) based on the collected token sets $\mathcal{S}_T$ and $\mathcal{S}_C$, which are denoted as *new-mono*. By using the state-tying technique, we can also train state-tied tri-phone HMMs for both positive and negative models from $\mathcal{S}_T$ and $\mathcal{S}_C$, which are denoted as *new-tri*.

In this experiment, we examine the capability of UV in terms of identifying word recognition errors from the recognition results of our baseline recognition system (with the ML models). We train all of our verification models only from the task-dependent training set *Train2*. The performance of UV is evaluated in both *Train2* and *Eva00*. We perform speech recognition for every utterances in *Train2* or *Eva00*. Then, a con-

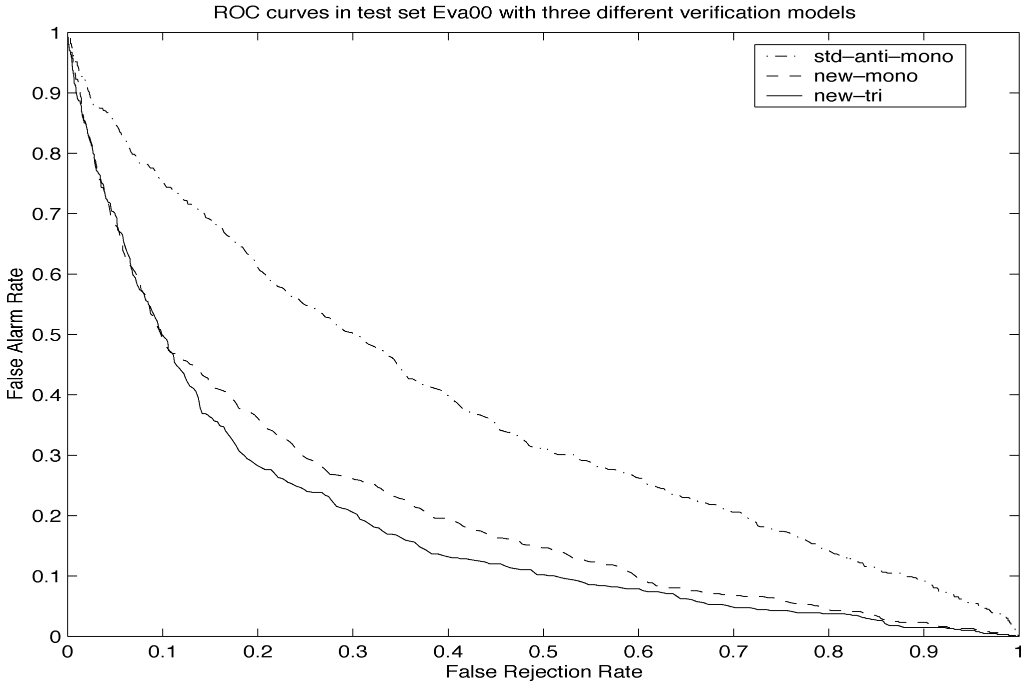

Fig. 3. Comparison of ROC curves in test data *Eva00* with three different verification models, namely *std-anti-mono*, *new-mono*, and *new-tri*, in verifying correct words versus misrecognized words.

fidence measure is calculated for every output word by combining the scores from all its phone segments. Based on the word-level confidence measures, we perform verification experiments between correctly recognized words versus misrecognized words (only including substitution and insertion errors). The performance comparison of EER (equal error rate) with different verification models is given in Table III. The results show that we can achieve EER 37.9% and 40.0% with the conventional anti monophone verification models in training set *Train2* (close test) and test set *Eva00* (open test), respectively. By using new mono-phone verification model *new-mono*, we can get EER 22.3% and 27.3% for *Train2* and *Eva00*. If we use tri-phone models, the performance can be further improved to 17.1% (for *Train2*) and 24.6% (for *Eva00*). As a reference, we also give the ROC curves with different models in test set *Eva00* in Fig. 3. All of these results clearly show that verification models trained with our new method significantly improve the verification performance in terms of identifying recognition errors, relatively 30% reduction in EER.

VII. Conclusion

As more and more speech data become available, it is more desirable to efficiently and intelligently use these training data to discover useful knowledge sources in designing a better and robust speech recognition system. Along this direction, we have proposed a data analysis procedure, where the recognition process based on Viterbi beam search is simulated to analyze training data in order to discover competing information automatically from speech data. As two examples, the collected competing information is used to improve acoustic modeling and utterance verification in ASR. Experimental results on DARPA communicator task clearly show that the proposed data analysis procedure can significantly improve the performance of both acoustic modeling and utterance verification. The preliminary study in this paper suggests that an effective and intelligent data analysis procedure, which can smartly determine the useful information pertinent to our particular purposes, is crucial to efficiently utilize a large amount of data and to build a better and more robust speech recognition system. Much more extensive studies are needed along this research direction.

References

[1] T. Anastasakos, J. McDonough, R. Schwarts, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. Int. Conf. Spoken Language Processing*, 1996, pp. 1137–1140.

[2] L. Arslan and J. Hansen, "Selective training for hidden Markov models with applications to speech classification," *IEEE Trans. Speech Audio Process.*, vol. 7, no. 1, pp. 46–54, Jan. 1999.

[3] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, Tokyo, Japan, 1986, pp. 49–52.

[4] M. DeGroot, *Optimal Statistical Decisions*. New York: McGraw-Hill, 1970.

[5] M. J. F. Gales, "Cluster adaptive training for speech recognition," in *Proc. Int. Conf. Spoken Language Processing*, Sydney, Australia, 1998, pp. 1783–1786.

[6] M. J. F. Gales, "Cluster adaptive training of hidden Markobv models," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 4, pp. 417–428, Jul. 2000.

[7] Y. Gao, B. Ramabhadran, and M. Picheny, "New adaptation techniques for large vocabulary continuous speech recognition," presented at the ISCA ITRW Workshop, Paris, France, Sep. 2000.

[8] J.-L. Gauvain and C.-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observation of Markov chains," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Mar. 1994.

[9] Q. Huo and C.-H. Lee, "On-line adaptive learning of the continuous density hidden Markov model based on approximate recursive Bayes estimate," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 2, pp. 161–172, Mar. 1997.

[10] Q. Huo and B. Ma, "Irrelevant variability normalization in learning structure from data: a case study on decision-tree based HMM state tying," in *Proc. ICASSP*, May 1999, pp. 577–580.

[11] H. Jiang and L. Deng, "A robust training strategy against extraneous acoustic variations for spontaneous speech recognition," in *Proc. 6th Int. Conf. Spoken Language Processing*, Beijing, China, Oct. 2000, pp. IV-161–IV-164.

[12] H. Jiang, F. Soong, and C.-H. Lee, "A data selection strategy for utterance verification in continuous speech recognition," in *Proc. EUROSPEECH*, Sep. 2001, pp. 2573–2576.

[13] H. Jiang and L. Deng, "A robust compensation strategy for extraneous acoustic variations in spontaneous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 1, pp. 9–17, Jan. 2002.

[14] H. Jiang, O. Siohan, F. Soong, and C.-H. Lee, "A dynamic in-search discriminative training approach for large vocabulary speech recognition," presented at the ICASSP, Orlando, FL, May 2002.

[15] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Commun.*, vol. 45, pp. 455–470, 2005.

[16] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error training," *IEEE Trans. Signal Process.*, vol. 40, no. 12, pp. 3043–3054, Dec. 1992.

[17] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 257–265, May 1997.

[18] C.-H. Lee, "A tutorial on speaker and speech verification," in *Proc. NORSIG*, Vigso, Denmark, Jun. 1998, pp. 9–16.

[19] R. Sukkar and C.-H. Lee, "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 6, pp. 420–429, Nov. 1996.

[20] 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. ICASSP*, Atlanta, GA, May 1996, pp. 516–519.

[21] R. Sukkar, A. R. Setlur, C.-H. Lee, and J. Jacob, "Verifying and correcting string hypotheses using discriminative utterance verification," *Speech Commun.*, vol. 22, pp. 333–342, 1997.

[22] M. G. Rahim, C.-H. Lee, and B.-H. Juang, "Discriminant utterance verification for connected digits recognition," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 3, pp. 266–277, May 1997.

[23] W. Reichl and W. Chou, "Robust decision tree state tying for continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 555–566, Sep. 2000.

[24] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in *Proc. ARPA Human Language Technology Workshop*, 1994, pp. 307–312.

[25] F. Wessel *et al.*, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 3, pp. 288–298, Mar. 2001.

[26] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 25–47, Jan. 2002.

Hui Jiang (M'00) received the B.Eng. and M.Eng. degrees from the University of Science and Technology of China (USTC) and the Ph.D. degree from the University of Tokyo, Tokyo, Japan, in September 1998, all in electrical engineering.

From October 1998 to April 1999, he was a Researcher at the University of Tokyo. From April 1999 to June 2000, he was with Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, as a Postdoctoral Fellow. From 2000 to 2002, he was with Dialogue Systems Research, Multimedia Communication Research Lab, Bell Laboratories, Lucent Technologies, Inc., Murray Hill, NJ. In 2002, he joined the Department of Computer Science, York University, Toronto, ON, as an Assistant Professor. His current research interests include all issues related to speech recognition and understanding, especially robust speech recognition, utterance verification, adaptive modeling of speech, spoken language systems, and speaker recognition/verification.

Frank K. Soong (SM'91) received the Ph.D. degree from Stanford University, Stanford, CA.

He joined Bell Laboratories Research, Murray Hill, NJ, in 1982 as a Member of Technical Staff and retired as a Distinguished Member of Technical Staff in 2001. After that, he spent two years, from 2002 to 2004, as an Invited Researcher at the Spoken Language Translation Laboratories, Advanced Telecommunication Research Institute (ATR), Kyoto, Japan. Over the years, he has worked on various aspects of speech research, including: speech and speaker recognition; speech segmentation, analysis and coding; stochastic modeling of speech signals; efficient search of multiple hypotheses via dynamic programming; discriminative training of HMMs; dereverberation of audio signals; microphone array signal processing; acoustic echo cancellation; and hands-free speech recognition in a noisy environment. He was responsible for transferring advanced speech recognition technology to AT voice-activated cell phones which were rated by the *Mobile Office Magazine* as the best among many competing products evaluated. His tree-trellis algorithm for finding the N-best sentence hypotheses forms the core of the popular free software JULIUS developed in Japan for speaker independent, large vocabulary, continuous speech recognition application. He has visited Japan twice as an Invited Researcher, first, from 1987 to 1988, at the NTT Electro-Communication Laboratories, Musashino, Tokyo, then from 2002 to 2004 at ATR. He is a Visiting Professor of the Chinese University of Hong Kong (CUHK) and the Co-Director of the CUHK-MSRA Joint Research Laboratory. He Co-Chaired the 1991 IEEE International Arden House Speech Recognition Workshop. He published extensively and has authored or coauthored more than 100 technical papers in the speech field.

Mr. Soong was the co-recipient of the Bell Laboratories President Gold Award for developing the Bell Laboratories Automatic Speech Recognition (BLASR) software package. He has served the IEEE Speech Technical Committee of the Signal Processing Society as a committee member and Associate Editor of the EEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. He Co-Chaired the 1991 IEEE International Arden House Speech Recognition Workshop.

Chin-Hui Lee (M'82–SM'91–F'97) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973, the M.S. degree in engineering and applied science from Yale University, New Haven, CT, in 1977, and the Ph.D. degree in electrical engineering, with a minor in statistics, from the University of Washington, Seattle, in 1981.

He joined Verbex Corporation, Bedford, MA, and was involved in research on connected word recognition. In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, CA, where he was engaged in research and product development in speech coding, speech synthesis, speech recognition, and signal processing for the development of the DSC-2000 Voice Server. Between 1986 and 2001, he was with Bell Laboratories, Murray Hill, NJ, where he became a Distinguished Member of Technical Staff and Director of the Dialogue Systems Research Department. From August 2001 to August 2002, he was a Visiting Professor with the School of Computing, The National University of Singapore. In September 2002, he joined the Faculty of School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, where he is currently a Professor. He has published more than 250 papers and holds 25 patents on the subject of automatic speech and speaker recognition. His research interests include multimedia communication, multimedia signal and information processing, speech and speaker recognition, speech and language modeling, spoken dialogue processing, adaptive and discriminative learning, biometric authentication, information retrieval, and bioinformatics. His research scope is reflected in *Automatic Speech and Speaker Recognition: Advanced Topics* (Norwell, MA: Kluwer, 1996).

Prof. Lee has participated actively in professional societies. He is a member of the European Speech Communication Association. He is also a lifetime member of the Computational Linguistics Society, Taiwan. From 1991 to 1995, he was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING. During the same period, he served as a member of the ARPA Spoken Language Coordination Committee. From 1995 to 1998, he was a member of the Speech Processing Technical Committee of the IEEE Signal Processing Society (SPS) and later became the Chairman of the Speech TC from 1997 to 1998. In 1996, he helped promote the SPS Multimedia Signal Processing (MMSP) Technical Committee, of which he is a founding member. He received the SPS Senior Award in 1994 and the SPS Best Paper Award in 1997 and 1999, respectively. In 1997, he was also awarded the prestigious Bell Laboratories President's Gold Award for his contributions to the Lucent Speech Processing Solutions product. In 2000, he was named one of the six Distinguished Lecturers by the IEEE Signal Processing Society.