Automatic Verbal Information Verification for User Authentication

Qi Li, *Member, IEEE*, Biing-Hwang Juang, *Fellow, IEEE*, Qiru Zhou, *Member, IEEE*, and Chin-Hui Lee, *Fellow, IEEE*

***Abstract*—Traditional speaker authentication focuses on speaker verification (SV) and speaker identification, which is accomplished by matching the speaker's voice with his or her registered speech patterns. In this paper, we propose a new technique, *verbal information verification* (VIV), in which spoken utterances of a claimed speaker are verified against the key (usually confidential) information in the speaker's registered profile automatically to decide whether the claimed identity should be accepted or rejected. Using the proposed sequential procedure involving three question-response turns, we achieved an error-free result in a telephone speaker authentication experiment with 100 speakers.**

We further propose a speaker authentication system by combining VIV with SV. In the system, a user is verified by VIV in the first four to five accesses, usually from different acoustic environments. During these uses, one of the key questions pertains to a pass-phrase for SV. The VIV system collects and verifies the pass-phrase utterance for use as training data for speaker model construction. After a speaker-dependent model is constructed, the system then migrates to SV. This approach avoids the inconvenience of a formal enrollment procedure, ensures the quality of the training data for SV, and mitigates the mismatch caused by different acoustic environments between training and testing. Experiments showed that the proposed system improved the SV performance by over 40% in equal-error rate compared to a conventional SV system.

***Index Terms*—Speaker authentication, speaker recognition, speaker verification, utterance verification, verbal information verification.**

I. Introduction

To ENSURE proper access to private information, personal transactions, and security of computer and communication networks, automatic user identification or authentication is necessary. Among various kinds of authentication methods, such as voice, password, personal identification number (PIN), signature, finger print, iris, hand shape, etc., voice is the most convenient one because it is easy to produce, capture, and transmit over the telephone or wireless network. It also can be supported with existing services without requiring special devices. Speaker recognition as one of the voice authentication techniques has been studied for several decades. There are however still several problems which affect real-world applications, such as acoustic mismatch, quality of the training data, the inconvenience of enrollment, and the creation of a large database to store all the enrolled speaker patterns. In order to enhance the speaker authentication technology, we present a novel approach called *verbal information verification* (VIV). It can be used independently or can be combined with speaker recognition to provide the convenience to users while achieving a higher level of security.

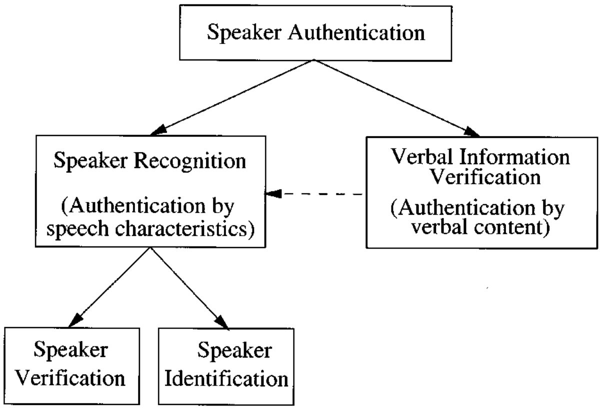

Speaker Authentication
Speaker Recognition
(Authentication by speech characteristics)
Verbal Information Verification
(Authentication by verbal content)
Speaker Verification
Speaker Identification

Fig. 1. Speaker authentication approaches.

To facilitate further discussion, we refer to *speaker authentication* as the general method of automatic authentication based on a speaker's voice input. It is the process of authenticating a user via his/her voice input using pre-stored information. The information can be in various formats, such as lexical transcriptions, acoustic models, text, subword sequences, etc. As shown in Fig. 1, speaker authentication can be categorized into two groups: by speaker's voice characteristics, as is done in conventional speaker recognition, or by the verbal content of an utterance, which leads to verbal information verification. Speaker recognition includes speaker verification (SV) and speaker identification (SID). SV is the process of verifying the claimed identity of an unknown speaker by comparing the speaker characteristics as encapsulated in spoken input against a pre-stored speaker-specific model. SID is the process of associating an unknown speaker with a member of a known population.

When applying the current speaker recognition technology to real-world applications, several problems were encountered which motivated our research of VIV [1], [2]. A conventional speaker recognition system needs an enrollment session to collect data for training a speaker-dependent (SD) model. Enrollment is an inconvenience to the user as well as the system developer who often has to supervise and ensure the quality of the collected data. The accuracy of the collected training data is critical to the performance of an SV system. Even a true

Manuscript received February 8, 1999; revised November 5, 1999. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Douglas O'Shaughnessy.

The authors are with the Multimedia Communications Research Lab, Bell Labs, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: qli@research.bell-labs.com; bjuang@research.bell-labs.com; qzhou@research.bell-labs.com; chl@research.bell-labs.com).

Publisher Item Identifier S 1063-6676(00)07086-3.

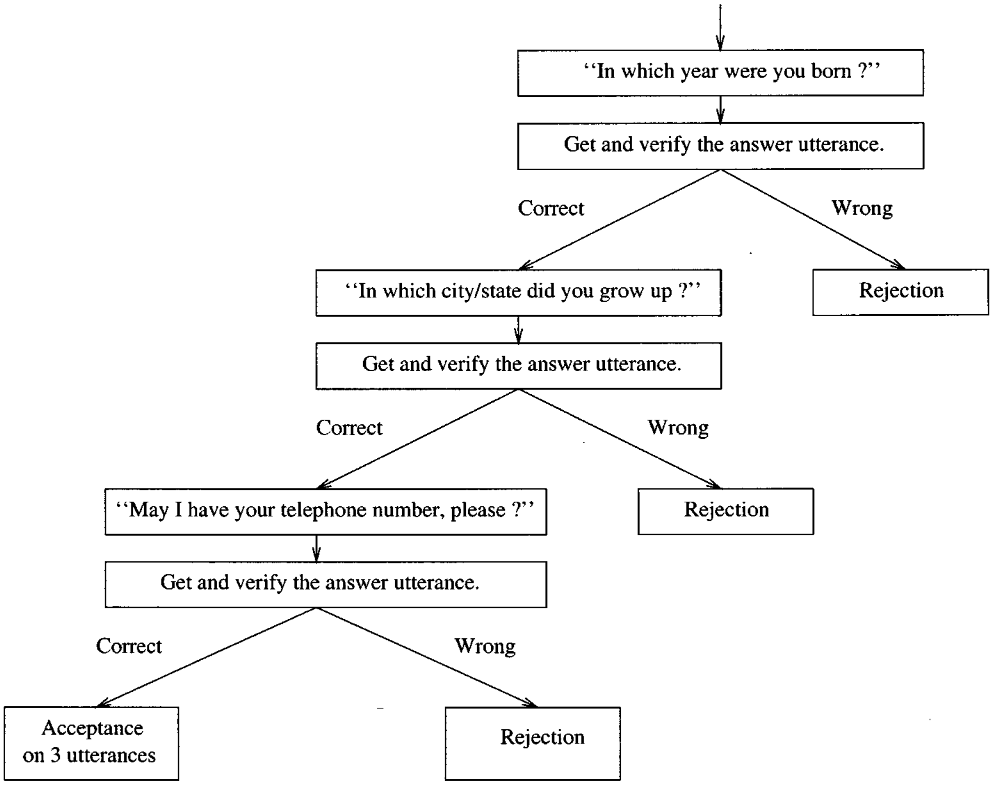

"In which year were you born ?"
Get and verify the answer utterance.
Correct
Wrong
"In which city/state did you grow up ?"
Rejection
Get and verify the answer utterance.
Correct
Wrong
"May I have your telephone number, please ?"
Rejection
Get and verify the answer utterance.
Correct
Wrong
Acceptance on 3 utterances
Rejection

Fig. 2. An example of verbal information verification by asking sequential questions. (Similar sequential tests can also be applied to speaker verification and other biometric or multi-modality verification.)

speaker might make a mistake when repeating the training utterances/pass-phrases for several times. Furthermore, as we discussed in [3], since the enrollment and testing voice may come from different telephone handsets and networks, there may exist an acoustic mismatch between the training and testing environments. The SD models trained on the data collected in one enrollment session may not perform well when the test session is in a different environment or via a different transmission channel. The mismatch significantly affects the SV performance. To alleviate the above problems, we proposed the concept and algorithm of VIV.

VIV is the process of verifying spoken utterances against the information stored in a given personal data profile. A VIV system may use a dialogue procedure to verify a user by asking questions. An example of a VIV system is shown in Fig. 2. It is similar to a typical tele-banking procedure: after an account number is provided, the operator verifies the user by asking some personal information, such as mother's maiden name, birth date, address, home telephone number, etc. The user must answer the questions correctly in order to gain access to his/her account. To automate the whole procedure, the questions can be prompted by a text-to-speech system (TTS) or by pre-recorded messages.

The major difference between speaker recognition and VIV in speaker authentication is that a speaker recognition system utilizes a speaker's voice characteristics represented by the speech feature vectors while a VIV system mainly inspects the verbal content in the speech signal.

The difference can be further addressed in the following three aspects. First, in a speaker recognition system, for either SID or SV, we need to train speaker-dependent (SD) models while in VIV, we usually use statistical models with associated acoustic-phonetic identities. Second, a speaker recognition system needs to enroll a new user and to train the SD model while a VIV system does not need such an enrollment. A user's personal data profile is created when the user's account is set up. Finally, in speaker recognition, the system has the ability to reject an imposter when the input utterance contains a legitimate pass-phrase but fails to match the pre-trained SD model. In VIV, it is solely the user's responsibility to protect his or her own personal information because no speaker-specific voice characteristics are used in the verification process. However, in real applications, there are several ways to avoid impostors using a speaker's personal information by monitoring a particular session. A VIV system can ask for some information that may not be a constant from one session to another, e.g., the amount or date of the last deposit; or a subset of the registered personal information, e.g., a VIV system can require a user to register N pieces of personal information ($N > 1$), and each time only randomly ask n questions ($1 \leq n < N$). Furthermore, as we are going to present in Section V, a VIV system can be migrated to an SV system as indicated by the dash line in Fig. 1. In particular, VIV can be used to facilitate automatic enrollment for SV.

The rest of the paper is organized as follows. In Section II, we present the algorithms of single utterance verification. In Section III, we propose a sequential utterance verification algorithm for VIV. Section IV gives the experimental results of VIV. Section V presents speaker authentication by combining VIV with

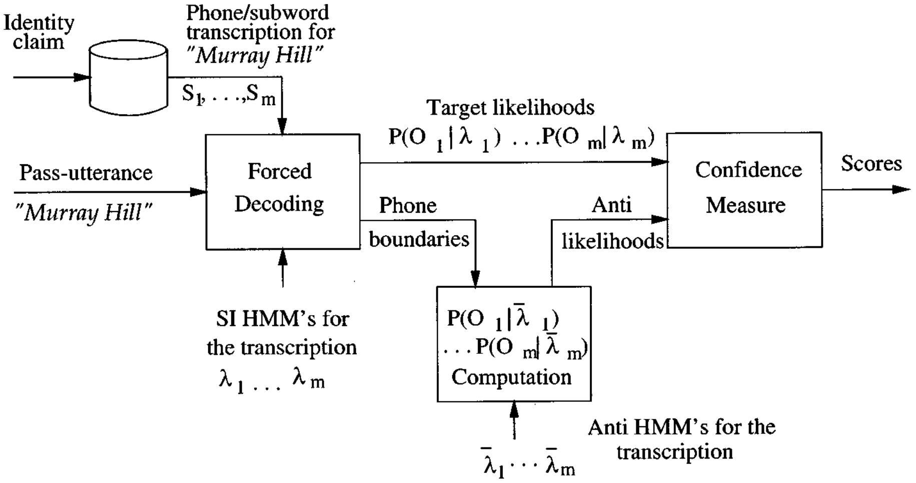


Fig. 3. Utterance verification in VIV.

SV and the experimental results, followed by the conclusions in Section VI.

II. Single Utterance Verification

There are two ways to verify a single spoken utterance for VIV: by automatic speech recognition (ASR) or by utterance verification. With ASR, the spoken input is transcribed into a sequence of words. The transcribed words are then compared to the information pre-stored in the claimed speaker's personal profile. With utterance verification, the spoken input is verified against an expected sequence of word or subword models which is taken from a personal data profile of the claimed individual. Based on our experience [1] and the analysis in Section III, the utterance verification approach can give us much better performance than the ASR approach. Therefore, we focus our discussion only on utterance verification approach in this study.

The idea of utterance verification for computing confidence scores was used in keyword sporting and nonkeyword rejection (e.g., [4]–[10]). A similar concept can also be found in fixed-phrase speaker verification [11], [12], [3] and in VIV [1], [2]. A block diagram of a typical utterance verification for VIV is shown in Fig. 3. The three key modules, utterance segmentation by forced decoding, subword testing and utterance level confidence scoring, are described in detail in the following subsections.

A. Utterance Segmentation

When a user opens an account, some of his or her key information is registered in a personal profile. Each piece of the key information is represented by a sequence of words, $\boldsymbol{S}$, which in turn is equivalently characterized by a concatenation of a sequence of phones or subwords, $\{S_n\}_{n=1}^{N}$, where S_n, is the nth subword and N is the total number of subwords in the key word sequence.

Since the VIV system only prompts one single question at a time, the system knows the expected key information to the prompted question and the corresponding subword sequence $\boldsymbol{S}$. We then apply the subword models $\lambda_1, \cdots, \lambda_N$ in the same order of the subword sequence $\boldsymbol{S}$ to decode the answer utterance. This process is known as forced decoding or forced alignment, in which the Viterbi algorithm is employed to determine the maximum likelihood segmentations of the subwords, i.e.

$$P(\boldsymbol{O}|\boldsymbol{S}) = \max_{t_1, t_2, \cdots, t_N} P\left(O_1^{t_1}|S_1\right) P\left(O_{t_1+1}^{t_2}|S_2\right) \cdots P\left(O_{t_{N-1}+1}^{t_N}|S_N\right) \tag{1}$$

where

$$\boldsymbol{O} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \cdots, \boldsymbol{O}_N\} = \left\{O_1^{t_1}, O_{t_1+1}^{t_2}, \cdots, O_{t_{N-1}+1}^{t_N}\right\} \tag{2}$$

is a set of segmented feature vectors associated with subwords, $t_1, t_2, \cdots, t_N$ are the end frame numbers of each subword segments respectively, and $\boldsymbol{O}_n = O_{t_{n-1}+1}^{t_n}$ is the segmented sequence of observations corresponding to subword S_n from frame number $t_{n-1}+1$ to frame number t_n, where $t_1 \geq 1$ and $t_i > t_{i-1}$.

B. Subword Hypothesis Testing

Given a decoded subword, S_n in an observed speech segment $\boldsymbol{O}_n$ we need a decision rule by which we assign the subword to either hypotheses H_0 or H_1. Following the definition in [8], H_0 means that observed speech $\boldsymbol{O}_n$ consists of the actual sound of subword S_n, and H_1, is the alternative hypothesis. For the binary-testing problem, one of the most useful test for decision is the *Neyman–Pearson* lemma [13]–[15]. For a given number of observations K, the most powerful test, which minimizes the error for one class while maintaining the error for the other class constant, is a likelihood ratio test,

$$r(\boldsymbol{O}_n) = \frac{P(\boldsymbol{O}_n|H_0)}{P(\boldsymbol{O}_n|H_1)} = \frac{P(\boldsymbol{O}_n|\lambda_n)}{P\left(\boldsymbol{O}_n|\overline{\lambda}_n\right)} \tag{3}$$

where λ_n and $\overline{\lambda}_n$ are the target HMM and corresponding anti-HMMs for subword unit S_n respectively. The target model, λ_n, is trained using the data of subword S_n; the corresponding anti-model, $\overline{\lambda}_n$ is trained using the data of a set of subwords $\overline{\boldsymbol{S}}$ which is highly confused with subword S_n [8], i.e., $\overline{\boldsymbol{S}}_n \subset \{S_i\}$, $i \neq n$. The log likelihood ratio (LLR) for subword S_n is

$$R(\boldsymbol{O}_n) = \log P(\boldsymbol{O}_n|\lambda_n) - \log P\left(\boldsymbol{O}_n|\overline{\lambda}_n\right). \tag{4}$$

For normalization, an average frame LLR, R_n, is defined as

$$R_n = \frac{1}{l_n}\left[\log P(\boldsymbol{O}_n|\lambda_n) - \log P\left(\boldsymbol{O}_n|\overline{\lambda}_n\right)\right] \tag{5}$$

where l_n is the length of the speech segment. For each subword, a decision can be made by

$$\begin{cases} \text{Acceptance:} & R_n \geq T_n; \\ \text{Rejection:} & R_n < T_n \end{cases} \tag{6}$$

where either a subword-dependent threshold value T_n or a common threshold T can be determined numerically or experimentally.

Here, we applied the concept of *Neyman–Pearson* lemma to utterance verification but we have to note that the lemma was not originally developed for HMM testing. Actually, the lemma is originally a test between two *pdfs*, which is equivalent to test the hypothesis in one HMM state. In the above test, we assume the independence among all HMM states and among all subwords. Therefore, the above test can be interpreted as applying the *Neyman–Pearson* lemma in every state, then combining the scores together as the final average LLR score.

C. Confidence Measures

For an utterance level decision, we have to define a function to combine the results of subword tests. A confidence measure $\mathcal{M}$ for a key utterance $\boldsymbol{O}$ can be represented as

$$\mathcal{M}(\boldsymbol{O}) = \mathcal{F}(R_1, R_2, \cdots, R_N) \tag{7}$$

where $\mathcal{F}$ is the function to combine the LLR's of all subwords in the key utterance.

Several confidence measures have been proposed for utterance verification (e.g., [6], [7]). We denote two of them as M_1 and M_2 in the following:

$$M_1 = \frac{1}{L}\sum_{n=1}^{N} l_n R_n \tag{8}$$

where N is the total number of nonsilence subwords in the utterance, and L is the total number of frames of the nonsilent portion of the utterance, i.e., $L = \sum_{n=1}^{N} l_n$. Furthermore,

$$M_2 = \frac{1}{N}\sum_{n=1}^{N} R_n. \tag{9}$$

Here, M_1, is an average score over all frames and all subwords. Each of the subword score R_n is weighted by its duration. M_2 is an average LLR of all subwords and independent of individual duration. We note that silence models are used in the forced alignment for utterance segmentation but only nonsilence subwords are involved in computing the confidence measures.

For VIV, we defined a different confidence measure, M, for two reasons. First, as reported in [6] and from our experiments, the above confidence measures have a large dynamic range. A preferable statistic should have a stable, limited numerical range, so that a common threshold can be determined for all subwords for simplicity. Second, decision thresholds should be determined to meet specifications in different applications. It is desirable to be able to relate the design specifications with the computed confidence measure. Again, the above confidence measures can not meet these requirements.

A useful design specification is the percentage of acceptable subwords in a key utterance. We then need to make a decision at both the subword and the utterance levels. At the subword level, a likelihood-ratio test can be conducted to reach a decision on acceptance or rejection of each subword; at the utterance level, a simple utterance score can be computed to represent the percentage of acceptable subwords.

To make a decision on the subword level, we need to determine the threshold for each of the subword tests. If we have the training data for each subword model and the corresponding anti-subword model, this is not a problem. However, in many cases, the data may not be available. Therefore, we need to define a test which can give us the convenience to determine the thresholds without using the training data. For subword S_n which is characterized by a model, λ_n, we define

$$C_n = \frac{\log P(\boldsymbol{O}_n|\lambda_n) - \log P\left(\boldsymbol{O}_n|\overline{\lambda}_n\right)}{\log P(\boldsymbol{O}_n|\overline{\lambda}_n)} \tag{10}$$

where $\log P(\boldsymbol{O}_n|\lambda_n) \neq 0$ means the target score is larger than the anti-score and vice versa. Furthermore, we define a *normalized confidence measure* for an utterance with N subwords as

$$M = \frac{1}{N}\sum_{n=1}^{N} f(C_n) \tag{11}$$

where

$$f(C_n) = \begin{cases} 1, & \text{if } C_n \geq \theta; \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

M is in a fixed range of $0 \leq M \leq 1$. Due to the normalization in (10), θ is a subword-independent threshold which can be determined separately. A subword is accepted and counted as part of the utterance confidence measure only if its C_n score is greater than or equal to the threshold value θ. Thus, M can be interpreted as the percentage of acceptable subwords in an utterance; e.g. $M = 0.8$ implies that 80% of the subwords in the utterance are acceptable. Therefore, an utterance threshold can be determined or adjusted based on the specifications of system performance and robustness.

III. Sequential Utterance Verification

The above single utterance test strategy can be extended to a sequence of subtests which is similar to the *step-down procedure* in statistics [16]. Each of the subtests is an independent single-utterance verification. As soon as a subtest calls for rejection, $\mathcal{H}_1$ is chosen and the procedure is terminated; if no subtest leads to rejection, $\mathcal{H}_0$ is accepted, i.e., the user is accepted.

Let $\mathcal{H}_0$ be the target hypothesis in which all the answered utterances match the key information in the profile. We have

$$\mathcal{H}_0 = \bigcap_{i=1}^{J} H_0(i) \tag{13}$$

where J is the total number of subtests, and $H_0(i)$ is a component target hypothesis in the ith subtest corresponding to the ith utterance. The alternative hypothesis is

$$\mathcal{H}_1 = \bigcup_{i=1}^{J} H_1(i) \tag{14}$$

where $H_1(i)$ is a component alternative hypothesis corresponding to the ith subtest. We assume the independence among subtests. On the ith subtest, a decision can be made on

$$\begin{cases} \text{Acceptance:} & M(i) \geq T(i); \\ \text{Rejection:} & M(i) < T(i) \end{cases} \tag{15}$$

where $M(i)$ and $T(i)$ are the confidence score and the corresponding threshold for utterance i, respectively.

As is well known, when performing a test, one may commit one of two types of error: rejecting the hypothesis when it is true—*false rejection* (FR), or accepting it when it is false—*false acceptance* (FA). We denote the FR and FA error rates as ε_r and ε_a, respectively. An *equal-error rate* (EER), ε, is defined when the two error rates are equal in a system, i.e., $\varepsilon_r = \varepsilon_a = \varepsilon$. For a sequential test, we extend the definitions of error rates as follows.

Definition 1: False rejection error on J utterances $(J \geq 1)$ is the error when the system rejects a correct response in any one of J hypothesis subtests.

Definition 2: False acceptance error on J utterances $(J \geq 1)$ is the error when the system accepts an incorrect set of responses after all of J hypothesis subtests.

Definition 3: Equal-error rate on J utterances is the rate at which the false rejection error rate and the false acceptance error rate on J utterances are equal.

We denote the above FR and FA error rates on J utterances as $E_r(J)$ and $E_a(J)$, respectively. Let $\Omega_i = \mathcal{R}_1(i) \cup \mathcal{R}_0(i)$ be the region of confidence scores of the ith subtest, where $\mathcal{R}_0(i)$ is the region of confidence scores which satisfy $M(i) \geq T(i)$ from which we accept $H_0(i)$, and $\mathcal{R}_1(i)$ is the region of scores which satisfy $M(i) < T(i)$ from which we accept $H_1(i)$.

The FR and FA errors for subtest i can be represented as the following conditional probabilities

$$\varepsilon_r(i) = P(M(i) \in \mathcal{R}_1(i) | H_0(i)), \tag{16}$$

and

$$\varepsilon_a(i) = P(M(i) \in \mathcal{R}_0(i) | H_1(i)) \tag{17}$$

respectively. Furthermore, the FR error on J utterances can be evaluated as

$$\begin{aligned} E_r(J) &= P\left(\bigcup_{i=1}^{J} \{M(i) \in \mathcal{R}_1(i)\} \middle| \mathcal{H}_0 \right) \\ &= 1 - \prod_{i=1}^{J} (1 - \varepsilon_r(i)) \end{aligned} \tag{18}$$

and the FA error on J utterances is

$$E_a(J) = P\left(\bigcap_{i=1}^{J} \{M(i) \in \mathcal{R}_0(i)\} \middle| \mathcal{H}_1 \right) = \prod_{i=1}^{J} \varepsilon_a(i). \tag{19}$$

Equations (18) and (19) indicate an important property of the sequential test defined above: the more the subtests, the less the FA error and the larger the FR error. Therefore, we can have the following strategy in a VIV system design: starting from the first subtest, we first set the threshold value such that the FR error rate for the subtest, ε_r, is close to zero or a small number corresponding to design specifications, then add more subtests in the same way until meeting the required system FA error rate, E_a, or reaching the maximum numbers of allowed subtests.

It is reasonable to arrange the subtests in the order of descending importance and decreasing subtest error rates. In other words, the system first prompts users with the most important question or with the subtest which we know has larger FR error $\varepsilon_r(i)$. Therefore, if a speaker is falsely rejected, the session can be restarted right away with little inconvenience to the user.

Equation (18) also indicates the reason that an ASR approach would not perform very well in a sequential test. Although ASR can give us low FR error, $\varepsilon_r(i)$, on each of the individual subtests, the overall FR error on J utterances $E_r(J)$, $J > 1$, can still be very high. In the proposed utterance verification approach, we make the FR on each individual subtest close to zero by adjusting the threshold value while controlling the overall FA error by adding more subtests until reaching the design specifications. We use the following examples to show the above concept.

Example 1: A bank operator usually asks two kinds of personal questions while verifying a customer. When automatic VIV is applied to the procedure, the average individual error rates on these two subtests are $\varepsilon_r(1) = 0.1\%$, $\varepsilon_a(1) = 5\%$; and $\varepsilon_r(2) = 0.2\%$, $\varepsilon_a(2) = 6\%$, respectively. Then, from (18) and (19), we know that the system FR and FA errors on a sequential test are $E_r(2) = 0.3\%$, and $E_a(2) = 0.3\%$. If the bank wants to further reduce the FA error, one additional subtest can be added to the sequential test. Suppose the additional subtest has $\varepsilon_r(3) = 0.3\%$ and $\varepsilon_a(3) = 7\%$. The overall system error rates will be $E_r(3) = 0.6\%$ and $E_a = 0.021\%$.

Example 2: A security system requires $E_r(J) \leq 0.03\%$ and $E_a(J) \leq 0.2\%$. It is known that each subtest can have $\varepsilon_r \leq 0.01\%$, and $\varepsilon_a \leq 12\%$ by adjusting the thresholds. In this case, we need to determine the number of subtests, J, to meet the design specifications. From (19), we have

$$J = \left\lceil \frac{\log E_a}{\log \varepsilon_a} \right\rceil = \left\lceil \frac{\log 0.002}{\log 0.12} \right\rceil = 3.$$

Then, the actual system FA rate on three subtests is $E_a = 0.17\% \leq 0.2\%$; the FR rate on three tests is $E_a(3) = 0.03\%$. Therefore, three subtests can meet the required performance on both FR and FA.

IV. VIV Experimental Results

In the following experiments, the VIV system verifies speakers by three sequential subtests, i.e., $J = 3$. The system performance with various decision thresholds will be evaluated and compared.

A. *Database, Features, and Models*

The experimental database includes 100 speakers. Each speaker gave three utterances as the answers to the following three questions:

"In which year were you born?"

"In which city and state did you grow up?" and

"May I have your telephone number, please?"

This is a biased database. Twenty six percent (26%) of the speakers have birth year in the 1950s; 24% are in the 1960s. There is only one digit different among those birth years. In city and state names, 39% are "New Jersey," and 5% of the speakers used exactly the same answer "Murray Hill, New Jersey." Thirty eight percent (38%) of the telephone numbers start from "908 582...," which means that at least 60% of the digits in their answer for the telephone number are identical. Also, some of the speakers have foreign accent, and some cities and states are in foreign countries.

In the experiments, a speaker was considered as a true speaker when the speaker's utterances were verified against his or her data profile. The same speaker was considered as an impostor when the utterances were verified against other speakers' profiles. Thus, for each true speaker, we have three utterances from the speaker and 99×3 utterances from other 99 speakers as impostors.

The speech signal was sampled at 8 kHz and pre-emphasized using a first-order filter with a coefficient of 0.97. The samples were blocked into overlapping frames of 30 ms in duration and updated at 10 ms intervals. Each frame was windowed with a Hamming window. The cepstrum was derived from a tenth order LPC analysis. The feature vector consisted of 39 features including 12 cepstral coefficients, 12 delta cepstral coefficients, 12 delta-delta cepstral coefficients, energy, delta energy, and delta-delta energy [17].

The models used in evaluating the subword verification scores were a set of 1117 right context-dependent HMMs as the target phone models [18], and a set of 41 context-independent anti-phone HMMs as anti-models [8].

B. *VIV by Sequential Utterance Verification*

In the following experiments, the sequential utterance verification approach presented in Sections II and III were evaluated.

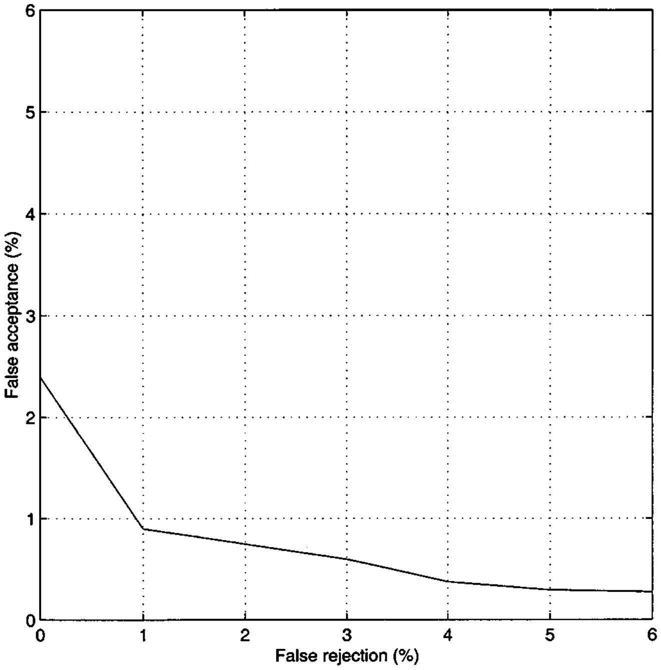


Fig. 4. Error rates when verifying three sequential utterances using a single SI threshold and confidence measure M.

All performances reported below are the overall performance on three questions as defined in Section III.

For a VIV system with multiple subtests, either one global threshold, i.e., $T = T(i)$, or multiple thresholds, i.e., $T(i) \neq T(j)$, $i \neq j$, can be used. The thresholds can be either context (key information) dependent or context independent. They can also be either speaker dependent or speaker independent. We discuss the threshold issues in the following.

1) Single Speaker-Independent Threshold: When a single SI threshold was applied to all speakers and all questions, the error rates in false rejection and false acceptance were obtained by varying the threshold value. As shown in Fig. 4, we have less than 1% equal-error rate when using confidence measure M defined in (11). Other confidence measures, e.g., M_2, can also give us less than 1% equal-error rates [1], but their thresholds can not be determined or adjusted based on design specifications, which is a basic requirement for the VIV system. Therefore, we focus our experiments on M only.

2) Two Speaker-Independent Thresholds: For robust sequential verification, we define the logic of using two speaker-independent and context-dependent thresholds for a multiple-question trial as follows:

$$T(i) = \begin{cases} T_L, & \text{when } T_L \leq M(i) < T_H \text{ at the first time} \\ T_H, & \text{otherwise} \end{cases} \tag{20}$$

where T_L and T_H are two threshold values; $M(i)$ and $T(i)$ are the values of confidence measure and threshold, respectively, for the ith subtest. Equation (20) means T_L can be used only once during the sequential trial. Thus, if a true speaker has only one lower score in a sequential test, the speaker still has the chance to pass the overall verification trial. This is useful in noisy environments or for speakers who may not speak consistently.

When the above two thresholds were applied to VIV testing, the system performance was improved from the single threshold test, as shown in Table I. The minimal false acceptance rates in the table were obtained by adjusting the thresholds while maintaining the false rejection rates to be 0%. As we can see from the table, the thresholds for M have limited ranges $0.0 \leq T_L$, $T_H \leq 1.0$ and clear physical meanings: i.e. $T_L = 0.69$ and $T_H = 0.84$ imply that 69% and 84% of phones are acceptable, respectively.

3) Robust Intervals: Due to the variations in voice characteristics, channels, and environment, a speaker may have various test scores, even for utterances of the same text. We define a *robust interval*, τ, to characterize the variation and the system robustness

$$\mathcal{T}(i) = T(i) - \tau, \qquad 0 \leq \tau < T(i) \tag{21}$$

where $T(i)$ is an original context-dependent utterance threshold as defined in (15), and $\mathcal{T}(i)$ is the adjusted threshold value. The robust interval, τ, is equivalent to the tolerance in the test score to accommodate fluctuation due to variations in environments or speaker's conditions.

In system evaluation, τ can be reported with error rates as an allowed tolerance; or it can be used to determine the thresholds based on system specifications. For example, a bank authentication system may need a smaller τ to ensure a lower false acceptance rate for a higher security level while a voice messaging system may select a larger τ for a lower false rejection rate to avoid user frustration.

4) Speaker-Dependent Thresholds: To further improve the performance, a VIV system can start from a speaker-independent threshold, then switch to speaker- and context-dependent thresholds after the system has been used for several times by a user.

To ensure no false rejection, the upper bound of the threshold for subtest i of a speaker can be selected as

$$T(i) \leq \min\{M(i, j)\}, \qquad j = 1, \cdots, I \tag{22}$$

where $M(i, j)$ is the confidence score for utterance i on the jth trial, and I is the total number of trials that the speaker has performed on the same context of utterance i.

In this case, we have three thresholds associated with the three questions for each speaker. Following the design strategy proposed in Section III, the thresholds were determined by first estimating $T(i)$ as in (22) to guarantee 0% false rejection rate. Then, the thresholds were shifted to evaluate the false acceptance rate on different robust intervals τ as defined in (21). The relation between robust interval and false acceptance rates on three questions using normalized confidence measure is shown in Fig. 5, where the horizontal axis indicates the changes of the values of robust interval τ. The three curves represent the performance of a VIV system using one to three questions for speaker authentication while maintaining false rejection rate to be 0%. An enlarged graph of the performance for the cases of two and three subtests is shown in Fig. 6. We note that the threshold adjustment is made on per-speaker, per-question situation although the plot in Fig. 6 is the overall performance for all speakers.

From the figures, we can see that using one question test, we cannot obtain a 0% equal-error rate. Using two questions, we have a 0% equal-error rate but with no tolerance (i.e., robust interval $\tau = 0$). With three questions, the VIV system gave 0% equal-error rate with 6% robust interval, which means when a true speaker's utterance scores are 6% lower than before (e.g., due to variations in telephone quality), the speaker can still be accepted while all impostors in the database can be rejected correctly. This robust interval gives room for variation in the true speaker's score to ensure robust performance of the system.

TABLE I
COMPARISON ON TWO AND SINGLE THRESHOLD TESTS

No. of SI thresholds	Error rates on three utterances	Threshold values
Two	FA = 0.57% FR = 0.0%	$T_L = 0.69$; $T_H = 0.84$
Single	FA = 0.75% FR = 1.0%	$T = 0.89$

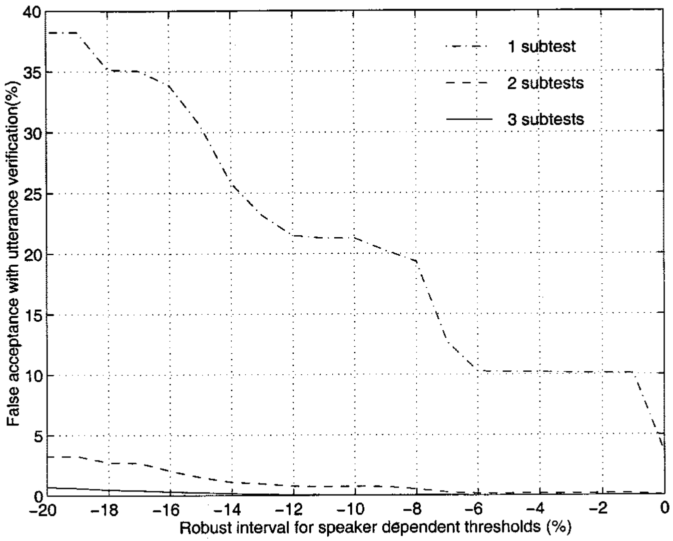


Fig. 5. False acceptance rate as a function of robust interval with SD threshold for a 0% false rejection rate. The horizontal axis indicates the shifts of the values of the robust interval τ.

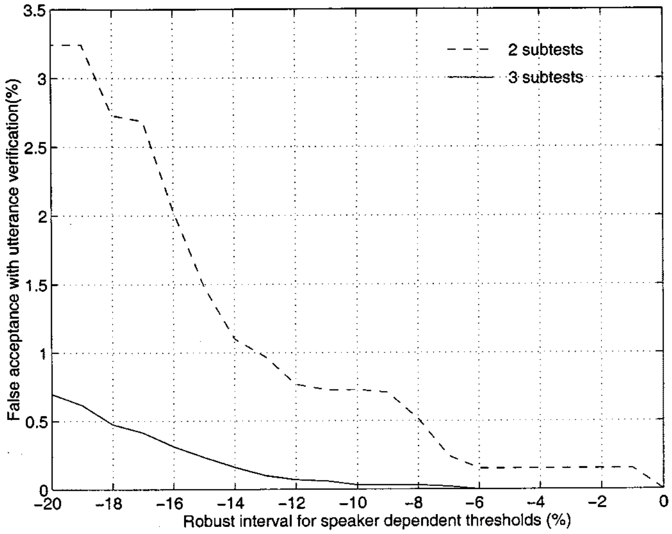


Fig. 6. An enlarged graph of the system performances using two and three questions.

TABLE II
SUMMARY OF THE EXPERIMENTAL RESULTS ON VERBAL INFORMATION VERIFICATION

Approaches	False Rejection	False Acceptance	Accuracy	Robust Interval
Sequential Utterance Verification	0%	0%	100%	6%

(Tested on 100 speakers with 3 questions while speaker-dependent thresholds were applied.)

Fig. 6 also implies that three questions are necessary to obtain a 0% false acceptance in the experiment.

A practical VIV system may apply SI thresholds to a new user and switch to SD thresholds after the user access the system successfully for a few times. The thresholds can also be updated based on the recent scores to accommodate the changes of speaker's voice and environment.

A summary of VIV for speaker authentication is shown in Table II. In the utterance verification approach, when SD thresholds are set for each key information field, we achieved 0% average individual equal-error rate with a 6% robust interval.

V. SPEAKER AUTHENTICATION BY COMBINING VIV WITH SPEAKER VERIFICATION

In the above sections, we have introduced VIV as an independent authentication method. In this section, we combine VIV with traditional speaker verification to construct a new speaker authentication system, which is more convenient to users with better performance by solving or mitigating the problems in enrollment, training data verification, and acoustic mismatch as discussed in Section I.

A. VIV for the Automatic Enrollment of Speaker Verification

A conventional SV system is shown in Fig. 7. It involves two kinds of sessions, enrollment and test. In an enrollment session, an identity, such as an account number, is assigned to a speaker, and the speaker is asked to select a spoken pass-phrase, e.g., a connected digit string or a phrase. The system then prompts the speaker to repeat the pass-phrase for several times, and an SD HMM is constructed based on the utterances collected in the enrollment session. In a test session, the speaker's test utterance is compared against the pre-trained, SD HMM model. The speaker is accepted if the likelihood-ratio score exceeds a preset threshold; otherwise the speaker is rejected.

The proposed approach [2] is shown in Fig. 8, where VIV is involved in the enrollment and one of the key utterances in VIV is the pass-phrase which will be used in SV later. During the first 4–5 accesses, the user is verified by a VIV system. The verified pass-phrase utterances are recorded and later used to train an SD HMM for SV. At this point, the authentication process can be switched from VIV to SV.

There are several advantages to the combined system. First, the approach is convenient to users since it does not need a formal enrollment session and a user can start to use the system right after his/her account is opened. Second, the acoustic mismatch problem is mitigated since the training data are from different sessions, potentially via different handsets and channels. Third, the quality of the training data are ensured since the training phrases are verified by VIV before establishing the SD HMM for the pass-phrase. Finally, once the system switches to SV, it would be difficult for an impostor to access the account even if the imposter knows the true speaker's pass-phrase.

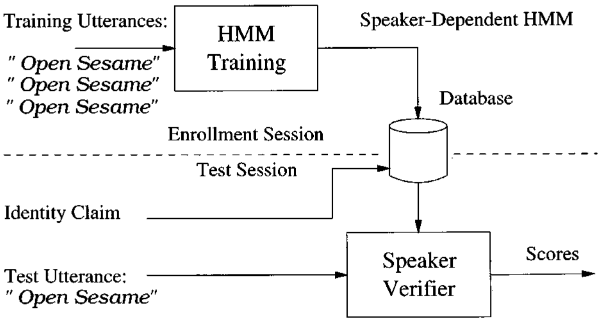


Fig. 7. Conventional speaker verification system.

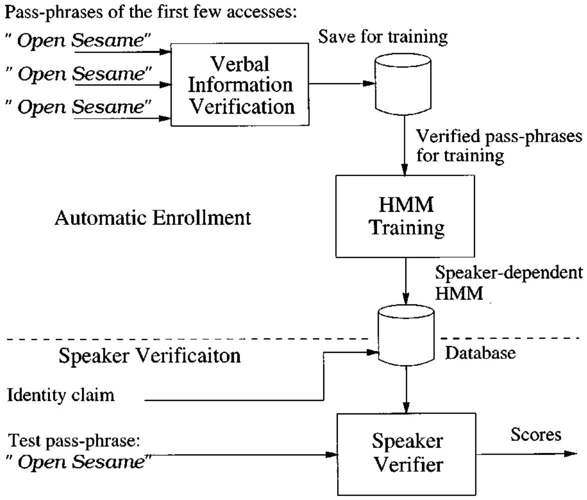


Fig. 8. Proposed system by combining VIV with speaker verification.

B. Fixed-Phrase Speaker Verification

The details of a fixed-phrase SV system can be found in [12]. A block diagram of the test session used in our evaluation is shown in Fig. 9. After the speaker claims the identity, the system expects the same phrase obtained in the training session. First, an SI phone recognizer is applied to find the end-points by forced alignment. Then, cepstral mean subtraction (CMS) is conducted to reduce the acoustic mismatch.

In the block of target score computation of Fig. 9, the feature vectors are decoded into states by the Viterbi algorithm, using the whole-phrase model trained by the VIV-verified utterances. A log-likelihood score for the target model, i.e. the target score, is calculated as

$$L(\boldsymbol{O}, \Lambda_t) = \frac{1}{N_f} \log P(\boldsymbol{O}|\Lambda_t) \tag{23}$$

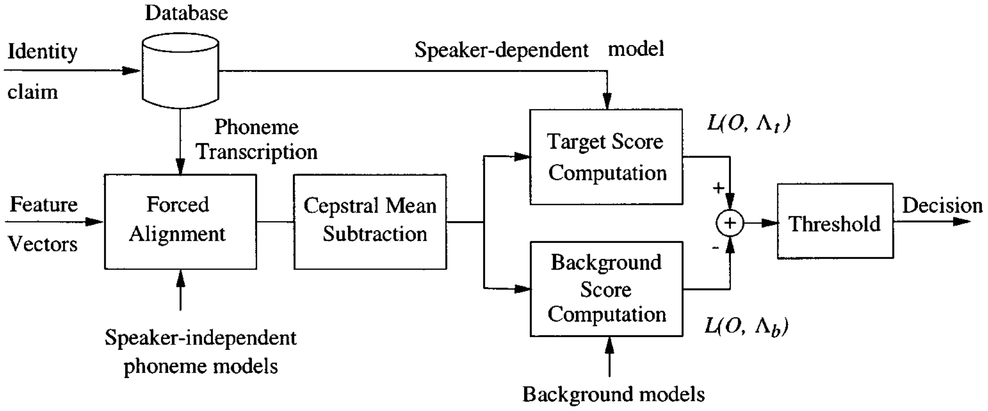


Fig. 9. Fixed-phrase speaker verification system.

where

O	set of feature vectors;
N_f	total number of vectors;
Λ_t	target model;
$P(O\|\Lambda_t)$	likelihood score from the Viterbi decoding.

In the block of the background score computation, a set of SI HMMs in the order of the transcribed phoneme sequence, $\Lambda_b = \{\lambda_1, \cdots, \lambda_K\}$, is applied to align an input utterance with the expected transcription using the Viterbi decoding algorithm. The segmented utterance is $O = \{O_1, \cdots, O_K\}$, where O_i is the set of feature vectors corresponding to the ith phoneme, S_i, in the phoneme sequence. There are different ways to compute the likelihood score for the background (alternative) model. Here, we apply the background score proposed in [12]

$$L(O, \Lambda_b) = \frac{1}{N_f} \sum_{i=1}^{K} \log P(O_i|\lambda_{b_i}) \tag{24}$$

where

$\Lambda_b = \{\lambda_{b_i}\}_{i=1}^{K}$	set of SI phoneme models, in the order of the transcribed phoneme sequence;
$P(O_i\|\lambda_{b_i})$	phoneme likelihood score;
K	total number of phonemes.

The SI models are trained from a different database by the EM algorithm [12]. In real implementation, the SI model can be the same one as used in VIV.

The target and background scores are then used in the following likelihood-ratio test [12]

$$\mathcal{R}(O; \Lambda_t, \Lambda_b) = L(O, \Lambda_t) - L(O, \Lambda_b) \tag{25}$$

where $L(O, \Lambda_t)$ and $L(O, \Lambda_b)$ are defined in (23) and (24) respectively.

A final decision on rejection or acceptance is made based on comparing $\mathcal{R}$ in (25) with a threshold. As pointed in [12], if a significantly different phrase is given, the phrase could be rejected by the SI phone alignment before using the verifier.

C. Features and Database

The feature vector for SV is composed of 12 cepstral and 12 delta-cepstral coefficients since it is not necessary to use the 39 features for SV. The cepstrum is derived from a 10th order LPC analysis over a 30 ms window and the feature vectors are updated at 10 ms intervals [12].

The experimental database consists of fixed phrase utterances recorded over the long distance telephone network by 100 speakers, 51 male and 49 female. The fixed phrase, common to all speakers, is “I pledge allegiance to the flag” with an average length of 2 seconds. We assume the fixed phrase is one of the verified utterances in VIV. Five utterances of the pass-phrase recorded from five separate VIV sessions are used to train an SD HMM, thus the training data are collected from different acoustic environments and telephone channels at different time. We assume all the collected utterances have been verified by VIV to ensure the quality of the training data.

For testing, we used 40 utterances recorded from a true speaker in different sessions, and 192 utterances recorded from 50 impostors of the same gender in different sessions. For model adaptation, the second, fourth, sixth, and eighth test utterances from the tested true speaker are used to update the associated HMM for verifying subsequent test utterances incrementally [12].

The SD target models for the phrases are left-to-right HMMs. The number of states are dependent on the total number of phones in the phrases. There are four Gaussian components associated with each state [12]. The background models are concatenated SI phone HMMs trained on a telephone speech database from different speakers and texts [11]. There are 43 phonemes HMMs and each model has three states with 32 Gaussian components associated with each state.

Due to unreliable variance estimates from a limited amount of speaker-specific training data, a global variance estimate was used as the common variance to all Gaussian components in the target models [12].

D. Experimental Results on Using VIV for SV Enrollment

In Section IV, we have reported the experimental results of VIV on 100 speakers. The system had 0% error rates when three questions were tested by sequential utterance verification. Therefore, we assume that all the training utterances collected by VIV are correct. Actually, since we are using a pre-verified database, we have to make the assumption. In other words, in the following experiment, we cannot show the improvement by ensuring the quality of the training data by VIV but the improvement by reducing the acoustic mismatch.

The SV experimental results without and with adaptation are listed in Tables III and IV for the 100 speakers respectively. The numbers are in the average percentage of individual equal-error rate (EER). The first data column lists the EERs using individual thresholds and the second data column lists the EERs using common (pooled) thresholds for all tested speakers.

The baseline system is the conventional SV system in which a single enrollment session is used. The proposed system is the combined system in which VIV is used for the automatic enrollment for SV. After the VIV system is used for five times by collecting training utterances from five different sessions, it then switches over to an SV system. The test utterances for both the baseline and the proposed system are the same.

Without adaptation, the baseline system has an EER of 3.03% and 4.96% for individual and pooled thresholds respectively, while the proposed system has an EER of 1.59% and 2.89%, respectively. With adaptation as defined in the last subsection, the baseline system has an EER of 2.15% and 3.12%, while the proposed system has an EER of 1.20% and 1.83%, respectively. The proposed system without adaptation has an even lower EER than the baseline system with adaptation. This is because the SD models in the proposed system were trained using the data from different sessions while the baseline system just performs an incremental adaptation without reconstructing the models after collecting more data.

The experimental results indicate several advantages of the proposed system. First, since VIV can provide the training data from different sessions representing different channel environments, we can do significantly better than one training session. Second, although we can adapt the models originally trained by the data collected in one session, the proposed system still does better. This is due to the fact that a new model constructed by multi-session training data is more accurate than by incremental adaptation using the multi-session data. Lastly, in real-world applications, all the utterances used in training and adaptation can be verified by VIV before training or adaptation. Although this advantage cannot be observed in this database evaluation, it is critical in any real-world application since even a true speaker may make a mistake while uttering a pass-phrase. The mistake will never be corrected once involved in model training or adaptation. VIV can protect the system from wrong training data.

In this section, we only proposed one configuration on combined VIV with SV. For different applications, different kinds of combinations and integration can be designed to meet different specifications. For example, VIV can be employed in SV to verify a user before the user's data is used for SD model adaptation; both the VIV and SV system can share the same set of SI models and the decoding scores from VIV can be used in SV as the background score; etc.

TABLE III
EXPERIMENTAL RESULTS WITHOUT ADAPTION IN AVERAGE EQUAL-ERROR RATES

Algorithms	Individual Thresholds	Pooled Thresholds
SV (Baseline)	3.03 %	4.96 %
VIV+SV(proposed)	1.59 %	2.89 %

TABLE IV
EXPERIMENTAL RESULTS WITH ADAPTION IN AVERAGE EQUAL-ERROR RATES

Algorithms	Individual Thresholds	Pooled Thresholds
SV (Baseline)	2.15 %	3.12 %
VIV+SV(proposed)	1.20 %	1.83 %

VI. CONCLUSIONS

In this paper, we presented automatic verbal information verification for user authentication. It is to verify speakers by verbal content instead of voice characteristics. We also proposed a sequential utterance verification solution to VIV with a system design procedure. Given the number of test utterances (subtests), the procedure can help us to design a system with minimal overall error rate; given a limit on the error rate, the procedure can find out how many subtests are needed to obtain the expected accuracy. In a VIV experiment with three questions prompted and tested sequentially, the proposed VIV system achieved 0% equal-error rate with 6% robust interval on 100 speakers when SD utterance thresholds were applied. However, since VIV is to verify the verbal content instead of the voice characteristics, it is users' responsibility to protect their personal information from impostors. The sequential verification technique can also be applied to other biometric verification systems, or multi-modality verification systems in which more than one verification methods can be employed, such as voice plus fingerprint verification, or other kinds of configurations.

To improve the user convenience and system performance, we further combined verbal information verification and speaker verification to construct a convenient speaker authentication system. In the system, VIV is used to verify users in the first few accesses. Simultaneously, the system collects verified training data for constructing SD models. Later, the system migrates to an SV system for authentication. The combined system is convenient to users since they can start to use the system without going through a formal enrollment session and waiting for model training. However, it is still the user's responsibility to protect his or her personal information from impostors until the SD model is trained and the system is migrated to an SV system. After the migration, an impostor would have difficulties to access the account event if the pass-phrase is known. On the other hand, since the training data could be collected from different channels in different VIV sessions, the acoustic mismatch problem is mitigated, potentially leading to a better system performance in test sessions. The SD HMMs can be updated to cover different acoustic environments while the system is in use to further improve the system performance. Our experiments showed that the combined speaker authentication system improved SV performance by more than 40% compared to that of a conventional SV system by just mitigating the acoustic mismatch. VIV can also be used to ensure training data for SV. Although the advantage cannot be shown in the experiments, it is critical to real-world applications.

ACKNOWLEDGMENT

The authors wish to thank S. Parthasarathy and A. E. Rosenberg for their original contributions to the fixed-phrase speaker verification system. Also, they would like to thank F. K. Soong and W. Chou for useful discussions.

REFERENCES

[1] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee, “Verbal information verification,” in *Proc. Eurospeech*, Rhode, Greece, Sept. 22–25, 1997, pp. 839–842.

[2] Q. Li and B.-H. Juang, “Speaker verification using verbal information verification for automatic enrollment,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, May 1998.

[3] Q. Li, S. Parthasarathy, and A. E. Rosenberg, “A fast algorithm for stochastic matching with application to robust speaker verification,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Munich, Germany, Apr. 1997, pp. 1543–1547.

[4] M. G. Rahim, C.-H. Lee, and B.-H. Juang, “Robust utterance verification for connected digits recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Detroit, MI, May 1995, pp. 285–288.

[5] M. G. Rahim, C.-H. Lee, B.-H. Juang, and W. Chou, “Discriminative utterance verification using minimum string verification error (MSVE) training,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. 3585–3588.

[6] T. Kawahara, C.-H. Lee, and B.-H. Juang, “Combining key-phrase detection and subword-based verification for flexible speech understanding,” in *Proc. Int. Conf. Acoustic, Speech, Signal Processing*, Munich, Germany, May 1997, pp. 1159–1162.

[7] E. Lleida and R. C. Rose, “Efficient decoding and training procedures for utterance verification in continuous speech recognition,” in *Proc. IEEE Int. Conf. Acoustics., Speech, Signal Processing*, Atlanta, GA, May 1996, pp. 507–510.

[8] R. A. Sukkar and C.-H. Lee, “Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition,” *IEEE Trans. Speech Audio Processing*, vol. 4, pp. 420–429, Nov. 1996.

[9] R. A. Sukkar, A. R. Setlur, M. G. Rahim, and C.-H. Lee, “Utterance verification of keyword string using word-based minimum verification error (WB-MVE) training,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. 518–521.

[10] A. R. Setlur, R. A. Sukkar, and J. Jacob, “Correcting recognition errors via discriminative utterance verification,” in *Proc. Int. Conf. Spoken Language Processing*, Philadelphia, PA, Oct. 1996, pp. 602–605.

[11] A. E. Rosenberg and S. Parthasarathy, “Speaker background models for connected digit password speaker verification,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. 81–84.

[12] S. Parthasarathy and A. E. Rosenberg, “General phrase speaker verification using sub-word background models and likelihood-ratio scoring,” in *Proc. Int. Conf. Speech Language Procsessing ’96*, Philadelphia, PA, Oct. 1996.

[13] J. Neyman and E. S. Pearson, “On the use and interpretation of certain test criteria for purpose of statistical inference,” *Biometrika*, pt. I, vol. 20A, pp. 175—240, 1928.

[14] ——, “On the problem of the most efficient tests of statistical hypotheses,” *Phil. Trans. R. Soc. A*, vol. 231, pp. 289–337, 1933.

[15] A. Wald, *Sequential Analysis*. London, U.K: Chapman & Hall, 1947.

[16] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 2nd ed. New York: Wiley, 1984.

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

[18] C.-H. Lee, B.-H. Juang, W. Chou, and J. J. Molina-Perez, “A study on task-independent subword selection and modeling for speech recognition,” in *Proc. Int. Conf. Speech Language Processing*, Philadelphia, PA, Oct. 1996, pp. 1816–1819.

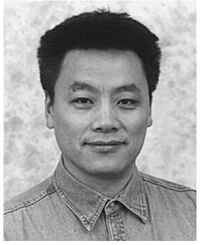
Qi Li (S’86–M’88) received the Ph.D. degree in electrical engineering from University of Rhode Island, Kingston.

From 1988 to 1994, he was with F.M. Engineering and Research, Norwood, MA, where he worked in research on patent recognition algorithms and in real-time systems. In 1991, he attended Harvard University to study statistical theory and methods. In 1995, he joined Bell Laboratories, Murray Hill, NJ, where is currently Member of Technical Staff in the Dialogue Systems Research Department. His research interests include speaker and speech recognition, fast search algorithms, stochastic modeling, robust features, fast discriminative learning, and neural networks. He has published regularly and holds patents in his research areas.

Dr. Li has been active as a reviewer for several journals, including IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING, and as a Local Chair for the IEEE Workshop on Automatic Identification. He has received two awards and is listed in *Who’s Who in America, Millennium Edition*.

Biing-Hwang Juang (S’79–M’80–SM’87–F’92) is Head of Acoustics & Speech Research Department, Bell Labs, Lucent Technologies, Murray Hill, NJ. He is engaged in a wide range of communication related research activities, from speech coding, speech recognition to multimedia communications. He has published extensively and holds a number of patents in the area of speech communication and communication services. He is co-author of the book *Fundamentals of Speech Recognition* (Englewood Cliffs, NJ: Prentice-Hall).

Dr. Juang received the 1993 Best Senior Paper Award, the 1994 Best Senior Paper Award, and the 1994 Best *Signal Processing Magazine* Paper Award, and was coauthor of a paper granted the 1994 Best Junior Paper Award, all from the IEEE Signal Processing Society. In 1997, he won the Bell Labs’ President Award for leading the Bell Labs Automatic Speech Recognition (BLASR) team. He also received the 1998 Technical Achievement Award from the IEEE Signal Processing Society and was named the Society’s 1999 Distinguished Lecturer. He was an Editor for the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING (1986–1988), the IEEE TRANSACTIONS ON NEURAL NETWORKS (1992–1993), and the *Journal of Speech Communication* (1992–1994). He has served on the Digital Signal Processing and the Speech Technical Committees as well as the Conference Board of the IEEE Signal Processing Society and was Chairman of the Technical Committee on Neural Networks for Signal Processing (1991–1993). He is currently Editor-in-Chief of the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING and is a member of the editorial board of the PROCEEDINGS OF THE IEEE. He also serves on international advisory boards outside the United States.



Qiru Zhou (S’86–M’93) received the B.S. and M.S. degrees from Northern Jiao-Tong University, China, and Beijing University of Posts and Telecommunications, China, respectively, in electrical and computer engineering.

He joined Bell Labs, AT&T, in 1992. He is now Member of Technical Staff at Bell Labs, Lucent Technologies, Murray Hill, NJ, working in the Dialogue Systems Research Department. His research interests include speech and speaker recognition algorithms and software, multimodal dialogue systems, real-time distributed object-oriented architecture for multimedia applications, and internet multimedia communications. Since 1992, he has been involved in various projects in AT&T and Lucent to apply speech technologies into products. Recently, he has been a Technical Leader in Lucent speech software product development.

Chin-Hui Lee (S'79–M'81–SM'90–F'97) received the B.S. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1973, the M.S. degree from Yale University, New Haven, CT, in 1977, and the Ph.D. degree from University of Washington, Seattle, in 1981, all in electrical engineering.

In 1981, he joined Verbex Corporation, Bedford, MA, and was involved in research work on connected word recognition. In 1984, he became affiliated with Digital Sound Corporation, Santa Barbara, CA, where he engaged in research in speech coding, speech recognition, and signal processing for the development of the DSC-2000 Voice Server. Since 1986, he has been with Bell Labs, Murray Hill, NJ, where he is now Distinguished Member of Technical Staff and the Head of Dialogue Systems Research Department at Bell Labs, Lucent Technologies. His current research interests include signal processing, speech modeling, adaptive and discriminative modeling, speech recognition, speaker recognition and spoken dialogue processing. He has published more than 200 papers in journals and international conferences and workshops on the topics in automatic speech and speaker recognition. His research scope is reflected in an edited book, entitled *Automatic Speech and Speaker Recognition: Advanced Topics* (Norwell, MA: Kluwer, 1996).

Dr. Lee was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and TRANSACTIONS ON SPEECH AND AUDIO PROCESSING from 1991 to 1995. He was a member of the ARPA Spoken Language Coordination Committee during the same period. Since 1995, he has been a member of the Speech Processing Technical Committee of the IEEE Signal Processing Society (SPS), in which he served as Chairman from 1996 to 1998. In 1996, he helped promote the newly formed SPS Multimedia Signal Processing (MMSP) Technical Committee and is now a member. He is a recipient of the 1994 SPS Senior Award and the 1997 and 1999 SPS Best Paper Award in Speech Processing. He was a winner of the prestigious Bell Laboratories President Gold Award in 1997 for his contributions to the Bell Labs Automatic Speech Recognition algorithms and products. Recently, he was chosen as one of the six distiguished lecturers of the Signal Processing Society for the year 2000.