A TUTORIAL ON SPEAKER AND SPEECH VERIFICATION

Chin-Hui Lee

Dialogue Systems Research Department
Bell Laboratories, Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ 07974, USA
chl@research.bell-labs.com

ABSTRACT

Speaker verification (SV) is the process of verifying the claimed identity of a speaker as one of the registered users based on his/her voice characteristics. *Utterance verification* (UV), on the other hand, attempts to verify the claimed content of a spoken utterance. When utterance verification is applied to verify the claimed identity of a talker against the stored information in his/her personal profile, the process is known as *verbal information verification* (VIV). In this paper, we discuss the fundamentals of pattern verification and focus on strategies and approaches that enable us to achieve speaker and speech verification for real-world applications.

1. INTRODUCTION

Applications of speaker and speech authentication can be found in access control, credit card authorization, and automatic teller machine user authorization. Based on different application requirements, conventional speaker authentication is classified into two major categories, namely *speaker identification* (SID) and *speaker verification* (SV). Speaker identification is the process of identifying an unknown speaker from a known population and speaker verification is the process of verifying the identity of a claimed speaker from a known population. In both SID and SV applications, an enrollment session is often required for the system to collect speaker-specific training data to build speaker models. Recently, a new technique, called *verbal information verification* (VIV) [7], is proposed to perform user authentication without the need of speaker enrollment. The idea is to verify a talker's claimed identity against the stored profile of the user, i.e. the system assumes the user's replies to the system's queries are known, just like the way it is usually performed by a human operator. *Utterance verification* (UV, e.g. [16]) is then used to verify if the user's spoken utterances match with the known content information. Different from the conventional speaker authentication scenarios, VIV can use speaker independent models to evaluate if the user has uttered the actual information. Therefore, any person who knows the requested information will potentially be authenticated. However if the personal information is assumed well-kept, the added flexibility of VIV enhances authentication to a level that almost perfect performance can be achieved [7] even some of the information provided by competing users can potentially be highly overlapping. VIV and SV can also be combined to further improve speaker authentication.

Although there are many ways to accomplish speaker and utterance verification, modern verification formulations are derived directly or indirectly from the theory of statistical hypothesis testing. In [5], a unified view on the fundamentals of verification is presented. It is argued that the conventional hypothesis testing theory is not directly applicable to designing optimal speaker and utterance verification tests due to the incomplete knowledge about the underlying speech distributions. The specific speech modeling framework, namely hidden Markov model (HMM), also calls for new theoretical justification about the conventional approaches. Another limitation of the existing theory is that model parameters in the conventional testing procedure need to be estimated from the training data. Due to the problem of insufficient training samples, optimal tests are even more difficult to design.

Conventional speaker training algorithms are based on *maximum likelihood* (ML) estimation of the model distribution and the parameters of the speaker models are estimated using only the training data from the same speaker. Recently, *discriminative training* algorithms [4, 8, 10] have been proposed to estimate model parameters based on the *minimum classification error* (MCE) and the *minimum verification error* (MVE) objectives. This MCE/MVE training approach takes into account other competing models and formulates the training criterion such that the recognition or verification error rate of the training data is directly minimized.

In this paper, we first review the fundamentals of pattern verification and formulate speaker and speech verification as a statistical hypothesis testing problem. SV, UV and VIV are the discussed. Finally an in-depth description of the MCE/MVE based training strategy is presented to highlight the importance of this new methodology. It is believed that this new paradigm can significantly change the way real-world verification problems are handled in the future.

2. PATTERN VERIFICATION

Verification of patterns is a problem we encounter in many situations. In general, the problem is formulated as follows: given a test signal X and some side information I, we want to verify if the signal X is generated from a signal source S_0. Two types of errors thus exist. First, one could have decided that X was *not* generated from the signal source, S_0, while it was indeed coming from the source. Second, one could have verified the given signal X as coming from the signal source S_0 while it was actu-

ally generated from a different source. The former, often referred to as *type I error*, is the error of *false rejection* or *missed detection*, and the latter, often referred to as *type II error*, is the error of *false acceptance* or *false alarm*. The verification performance is often evaluated as a combination of Type I and Type II errors. It usually depends on the source distribution, $P(X|S_0)$ and the amount of information contained in the test signal, X. However, it is also important to know the properties of competing source distributions in order to optimize the performance.

2.1. Statistical Hypothesis Testing

Based on our knowledge in statistics, the above pattern verification scenario is often conveniently formulated as a *hypothesis testing* problem: given the test signal X, we want to test the *null hypothesis*, H_0, against the *alternative hypothesis*, H_1, where H_0 assumes that X is generated from the source, S_0, and H_1 assumes that X is generated from another source, S_1. In many of the general speaker and utterance verification problems, H_1 assumes that X is *not* generated from the known source, S_0, which means H_1 is a *composite hypothesis* as opposed to being a *simple hypothesis* which is the case commonly dealt with in textbooks. This makes designing optimal tests even more involved. In principle, a test procedures tried to divide the signal space S_X into two regions W_X and $U_X = S_X - W_X$ and decide to reject H_0 if it is found that $X \in W_X$ and accepts H_0 if $X \in U_X$. W_X is often referred to as the *critical region* of the test. Now the probability of the two types of errors, E_1 and E_2, can be computed as

$$\alpha = P(E_1) = P(X \in W_X|H_0), \tag{1}$$

and

$$\beta = P(E_2) = P(X \in U_X|H_1) = 1 - P(X \in W_X|H_1). \tag{2}$$

The *power* of the test, which is an important quality in some textbooks, is computed as

$$\gamma = P(X \in W_X|H_1). \tag{3}$$

In statistical hypothesis testing, one is often interested in finding the critical region W_X such that the power of the testing procedure is maximized, or in other words, the type II error is minimized, at a given level of type I error. This is often referred to as finding the *most powerful test*. There are plenty of techniques available in literature in designing optimal tests if $P(X|H_0)$ and $P(X|H_1)$ are known exactly and they fall into a specific class of distributions, such as the *exponential family* (e.g [6]). However, for most of the practical verification problems, we have no exact knowledge about the distributions of the null and alternative hypotheses. The problem is even more involved in speech modeling because the test signal, X, is often a dynamic pattern which is nonstationary. The parameters needed are usually estimated using a given set of speech examples that was generated from some of the known sources. Due to model incorrectness and estimation errors, there is no theory or testing procedure that is optimal in the classical sense.

Based on the formulation we presented above, we perform pattern verification as follows: given a test pattern, X, we form a test statistic, $T(X)$, and accept H_0 if

$$T(X|H_0) \geq \omega, \tag{4}$$

where ω is a test threshold.

2.2. Pattern Verification Evaluation Issues

Different from identification or recognition problems in which the misrecognition error is the typical performance metric, we now have two kinds of errors, namely false rejection and false acceptance rates, to be considered. These two types of error are determined by the verification threshold used, and the two distributions of the test statistic, one for the null and the other for the alternative hypothesis. A typical plot of the two competing distributions is shown in Figure 1, where the distributions for the null and alternative hypotheses are shown in the right and the left of the plot respectively. The threshold value is indicated by the vertical line shown between the two distributions. The shaded area on the right, referred to as region II in Figure 1, denoted as $R_{II} = \{X|T(X) \geq \omega \ and \ X \in H_1\}$, indicates the region of false acceptance. On the other hand, the shaded area on the left, referred to as region I in Figure 1, denoted as $R_I = \{X|T(X) \leq \omega \ and \ X \in H_0\}$, indicates the region of false rejection. The threshold value that makes the areas of R_I and R_{II} equal is the *equal error* threshold. A test usually performs well if the two distributions are far apart from each other. Part of the verification design is to find a test and its corresponding parameters so that the overlap between the two distributions is minimized.

Since the overlap in Figure 1 always exists for real-world verification applications, the choice of the threshold value is crucial in determining the tradeoff between the false rejection and false acceptance errors. We can plot the two types of errors and their sums of errors as a function of the threshold value. Another way of showing the tradeoff is to plot the *receiver operating characteristic* (ROC) curves of the test. Borrowing from detection theory, the ROC curve is a plot of the detection rate versus the false alarm rate. It is also a good tool to compare performance of different test procedures by examining the corresponding ROC curves on the same figure.

2.3. Probability and Likelihood Ratio Tests

Since *probability ratio test* (PRT), or *likelihood ratio test* (LRT), is known to perform well and shown optimal by the classical *Neymann-Pearson Lemma* (e.g [6]) for many known distributions, researchers in the speech community

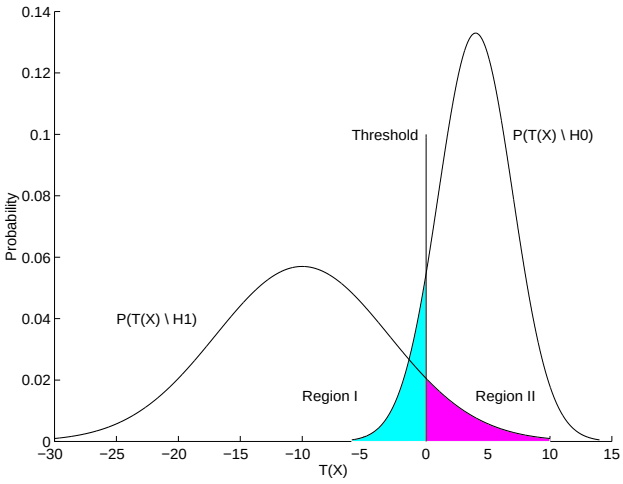

Figure 1: Distributions of test statistic under H_0 and H_1

have recently adopted the likelihood ratio test as a way to perform speaker and utterance verification, i.e. based on the *likelihood ratio statistic*, $T(X)$, the verification test accepts H_0 if

$$T(X) = f(X|\lambda_0)/f(X|\lambda_1) \geq \omega^* \qquad (5)$$

where ω^* is the *threshold* of the test, λ_0 and λ_1 are model parameters characterizing H_0 and H_1 respectively, and $f(X|\lambda_0)$ and $f(X|\lambda_1)$ are the likelihoods that the test signal, X, is generated by the two competing sources, H_0 and H_1, respectively.

The difference in Eq. (5) from the conventional, *unnormalized* likelihood test in Eq. (4) lies in the inclusion of the term $f(X|\lambda_1)$ in the evaluation of the test score. This makes the verification test closely linked with the statistical hypothesis testing formulation discussed in Section 2.1 that a better characterization of the alternative hypothesis H_1 improves performance of the test. Eq. (5) is known as a *normalized score function* for speaker verification [8]. Most recent utterance verification algorithms are based on this formulation [10, 16].

2.4. Speaker and Utterance Verification

Based on the statistical hypothesis testing formulation, speaker verification can be accomplished as follows: given a test utterance, X, from a talker with a claimed identity, q, we want to test the null hypothesis, H_0, that X is given by speaker q, against the alternative hypothesis, H_1, that X is *not* generated by speaker q. Usually, we form a test statistic, $T(X)$, and accept H_0 if $T(X|\lambda_q) \geq \omega_q$, where ω_q is a test threshold and λ_q is a model for speaker q.

Similarly, utterance verification is formulated as follows: given a segment of test utterance, X, with a claim content of linguistic information, W, we want to test the null hypothesis, H_0, that X contains W, against the alternative hypothesis, H_1, that X does *not* contains W. For verbal information verification, which is an application of utterance verification, W is usually a word, a phrase, or a sentence that characterizes the personal information, such as *mother's maiden name*, *user's birthday and birth place*, or *when was the user's last transaction and how was the transaction*. Again, we usually form a test statistic $T(X)$, and accept H_0 if $T(X|\lambda_W) \geq \omega_W$, ω_W being a test threshold and λ_W being a model for the linguistic event W.

It is clear the above two verification problems are similar in nature. It will also be seen in the following sections that the speaker q and the linguistic event W are often modeled using the *hidden Markov modeling* framework (e.g. [9]) to capture both the spectral and temporal variations in the speech signal.

It is worth noting the relationship between pattern verification and recognition. For instance, in speaker identification, we are interested in identifying the speaker as one of the registered users in a population. On the other hand, in speech recognition, we are interested in identifying the linguistic content embedded in a spoken utterance. Although the theory behind verification and recognition is based on the Neymann-Person Lemma and the *Bayes classification* respectively, the speech modeling and dynamic programming search techniques used in speaker and utterance verification are mostly borrowed from those used in speech recognition. By taking advantage of the advances made in speech recognition, our capabilities in speech pattern verification have been greatly enhanced.

3. SPEAKER VERIFICATION

As discussed earlier, speaker verification is the process of verifying the identity claim of a talker based upon his or her spoken utterances. A typical speaker verification is shown in Figure 2. Given a sequence of speech feature vectors, X, and the claimed identity, q, the test score, $T(X|\lambda_q)$, is computed based on the corresponding speaker models, λ_q. The score is then compared with the threshold associated with the claimed speaker, ω_q, to decide if the claimed identity should be accepted or rejected.

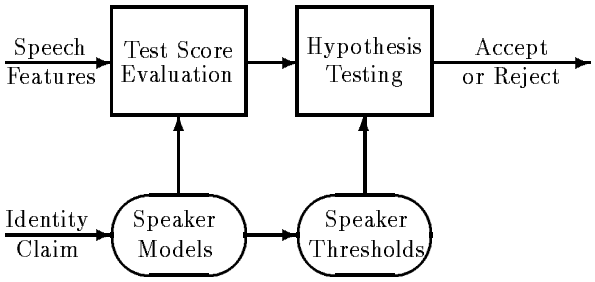

Figure 2: A typical speaker verification system

It is obvious that speaker modeling is the most crucial part in designing an effective speaker verification system. A *direct* way of modeling a speaker is to use the physiological parameters of the speaker's speech production system, such as the vocal tract size, to characterize the speaker. Since it is not easy to uniquely extract such production parameters from the given speech signal, X, *direct speaker modeling* has not made much progress lately. Instead, *indirect speaker modeling* is often used. It creates speech models from a given set of training samples from the speaker. Such a collection of speech models is then used as the speaker model for the particular speaker. The modeling technique varies depending on the desired verification strategy. We will talk about these strategies first before discussing the important issue of indirect speaker modeling through speaker-specific speech modeling.

3.1. Speaker Verification Strategies

There are roughly four types of strategies, namely *fixed phrase verification*, *fixed vocabulary verification*, *flexible vocabulary verification*, and *text-independent verification*. For fixed phrase verification, a pre-determined phrase is used to collect training utterances from a speaker to create speaker models and the speaker uses the same fixed phrase to generate test utterances. Many traditional *dynamic time warping* (DTW) based systems use this strategy.

To make verification more flexible and practical, a fixed-vocabulary verification strategy can be employed. All the training and testing materials for a speaker are generated based on some or all the words of a fixed vocabulary, The digit is probably the most commonly used fixed vocabulary in many languages because its size is relatively small and many *personal identification numbers* (PINs) are composed of sequences of digits. When digits are used for pass phrases, they can either be numbers known to the user or unknown numbers generated

randomly and prompted at the time of testing [2]. This *prompt verification* strategy is designed to prevent the impostors from recording a user's pass phrases and playing them back in an attempt to break the systems based on fixed pass phrases.

In contrast to the fixed vocabulary strategy, we can adopt flexible vocabulary verification in which a general set of subword phone models is created during speaker model training. Along with a given lexicon, any word or phrase is a particular language can then be modeled. Therefore any fixed or prompted strings of words can potentially be used as pass phrases for speaker verification.

The most general verification strategy is the so called text-independent verification in which the user is not constrained to say fixed or prompted phrases. Instead, the user is free to utter anything during testing. Since the text and vocabulary information corresponding to the spoken utterances is not used during speaker training and testing, there is no need to label the training and testing data. Phonetic information is also not needed in phone modeling and word and phrase composition. Given the same amount of testing data, the text-independent strategy usually does not work as well compared with the text-dependent ones. Therefore in addition to conventional SV, a new mode of verification,called *speaker monitoring*, which does not require the use of lexical information is often considered. Here the system operates in a *sequential hypothesis testing* mode, i.e. the user is constantly checked to be either accepted, rejected or more testing speech be collected for future verification decisions.

In all the above four strategies, speaker modeling needs to be done first during a training stage, called *speaker enrollment*. We now discuss the crucial issue of indirect speaker modeling.

3.2. Indirect Speaker Modeling

From the above discussion it is clear that different speech modeling techniques are needed depending on the type of strategy used. For example, in text-independent verification, any reasonable speech modeling algorithms can be used because no lexical information is required.

The simplest way to model speech is through *vector quantization* (VQ) in which a VQ codebook with C codewords, $B_q = \{c_{qi}, i = 1, , , C\}$ is created for each speaker, the test then accepts the identity claim if the total utterance distortion, $D(X, B_q) = \sum_{t=1}^{T} \min_{b \in B_q} d(x_t, b)$ is below a threshold (e.g. [1, 15]) with $d(x_t, b)$ being a frame distortion. To be consistent with our earlier notation, the test statistic is defined as, $T(X|B_q) = -D(X, B_q)$. By extending from memoryless VQ to *matrix quantization*, temporal constraints in speech can also be incorporated to improve verification performance.

Another way of modeling speech is to assume the speech frames are generated from a stochastic source with a *Gaussian mixture model* (GMM, e.g. [11]), i.e. the test statistic is based on the following likelihood,

$$T(X|\lambda_q) = \prod_{t=1}^{T} \sum_{m=1}^{M} w_{qm} \mathcal{N}(x_t|\mu_{qm}, \Sigma_{qm}), \qquad (6)$$

where M is the number of mixture components, w_{qm}, μ_{qm} and Σ_{qm} are the mixture gain, mean vector and covariance matrix of the mth mixture component respectively, and $\mathcal{N}(\cdot)$ is the Gaussian probability density function.

By incorporating the temporal structure into speech modeling, the speaker can further be modeled by an HMM, such as an ergodic HMM [18]. When 5 states are used in speech modeling, it was shown that the states correspond to roughly the five broad acoustic phonetic classes in spoken English [18], each can then be characterized by a mixture Gaussian density.

Instead of using only five states, a set of *acoustic segment models* can also be used to model a speaker [13]. Here one can imagine each acoustic segment model corresponds to a sound class and it is modeled by a left-to-right HMM so as to capture the temporal variation in speech.

When the lexical information is available, i.e. the speech training utterances are labeled with vocabulary information, text-specific or vocabulary-dependent speech modeling can be used. This is where acoustic modeling techniques in speech recognition can be widely applied.

The HMM formulation has been successfully applied to a number of speech and speaker recognition problems (e.g. [8, 9, 13]). For the purposes of speaker recognition, let us assume that each speaker q is modeled by a set of HMMs $\Lambda_q = \{\lambda_{qr}, 1 \leq r \leq R\}$ in which λ_{qr} denotes the HMM for the rth speech unit of speaker q. The set of speech units characterizing a speaker's acoustic space can either be a codebook of vector quantized codewords [15], a set of acoustic units [18], a collection of acoustic segment models [13], a set of subword phone units [13], a set of whole word (typically digit) units [14], or a single text-specific utterance unit. Many HMM training algorithms have been successfully applied to training speech models. However, ML training is still the most often used training algorithm. We will give a detailed account of the discriminative training strategies in Section 6.

3.3. Anti-Speaker Modeling

Although the the conventional verification formulation of Eq. (4) has been used extensively in the past, we can still ask the questions: "How good is the test judging from the theory of hypothesis testing ?" and "How can be we better if it is not good enough ?" In designing a verification system, one needs not only to create a set of accurate speaker models but also to determine the verification thresholds related to these models. In the verification test shown in Eq. (4) the threshold ω_0 is chosen independent of the testing utterance X and the alternative hypothesis H_1, which is often a *composite hypothesis*. However, according to the Neymann-Pearson Lemma (e.g. [6]), such a *uniformly most powerful test* (UMPT), as shown in in Eq. (4), can not always be realized because: (1) We do not know the form of the distributions of X. Even we assume that X is generated by an HMM, we still don't know how to design a UMPT; (2) Even we know the form of the distributions, we still have to estimate the parameters of the distributions because they are not known exactly in most practical situations; and (3) H_1 is a composite hypothesis. Since no existing theory is directly applicable to design such a test optimally, one can adopt the notion that a test of the form of a *generalized likelihood ratio test*, similar to the one in Eq. (4),

$$T(X|\lambda_q, \lambda_{\bar{q}}) = \frac{\mathcal{L}(X|\lambda_q)}{f(\mathcal{L}(X|\lambda_{\bar{q}}))} \geq \omega_q \qquad (7)$$

be used. Here $\lambda_{\bar{q}}$ is a collection of models to characterize H_1 in testing H_0, and $f(\cdot)$ is a function of the likelihoods of these competing models in H_1.

One way to improve our understanding why Eq. (7) is better than Eq. (4) is to look only into the subspace of speaker models that is most close to the claimed speaker, i.e. the most competitive to the claimed speaker. Thus uss analogous to the so called *locally most powerful test* (LMPT) in statistics literature (e.g. [6]). We can then simplify H_1 by assuming it to be a collection of simple hypotheses. This is the main idea behind the successful usage of "*cohort*" (e.g. [3, 14]). The cohort set, C_q is usually the set of most competitive speakers for the claimed speaker q. They are determined in the enrollment stage from the training speakers in the population. In [14], it was found that $f(\mathcal{L}(X|\lambda_{\bar{q}})) = \max_{r \in C_q} \mathcal{L}(X|\lambda_r)$ gives the best verification performance. This is intuitively appealing. However, there are cases in which more than one competing speaker is needed. The maximization operation is also not directly embedded into a functional optimization problem as will be dealt with in Section 6.

To circumvent the above difficulty, one can adopt an L_η-norm approximation [4, 8],

$$f(\mathcal{L}(X|\lambda_{\bar{q}})) = \ln\{\frac{1}{|C_q|}\sum_r \exp[\eta \ln \mathcal{L}(X|\lambda_r)]\}^{\frac{1}{\eta}}, \quad (8)$$

where $|C_q|$ is the size of the cohort set and η is a positive constant. It is clear that all the likelihoods of the cohort speakers contribute to the evaluation of the test statistic. It is also clear that as $\eta \to \infty$, the quantity in Eq. (8) converges to $\max_{r \in C_q} \ln \mathcal{L}(X|\lambda_r)$, a desirable property as mentioned above. When including Eq. (8) in the denominator of Eq. (7), it was found that based on single digit testing, the speaker verification equal error was reduced from 6.1% to 1.1% [8], an 80% error reduction. We will revisit this important concept again in Section 6.

There is actually no need to restrict our thinking to only cohort. one can always group together all the enrollment data of a speech unit from all the cohort speakers and build an *anti-model* for the particular speech unit. The collection of such anti-models forms an *anti-speaker* model, denoted again by $\lambda_{\bar{q}}$. One can also construct a "general" speech anti-model by using a speaker independent model. One can also make it independent of the speech unit by using a universal HMM sink model (e.g. [19]), or abandon the notion of HMM states and simply use a general mixture Gaussian anti-speaker model [11].

4. UTTERANCE VERIFICATION

We now turn our attention to the relatively new area of utterance verification which is the process of verifying the content of a segment of speech. Since the content here can be attached to a user's personal information profile, it also serves as a link for identifying a speaker. Instead of matching the speaker's voice with a set of pre-enrolled, speaker-dependent speech models as in SV, we now match the speaker's utterances with speech models that correspond to some pre-stored text information. Using a set of speaker-independent phone models and a lexicon, any text pass phrase can be modeled in principle without going through speaker enrollment. This new speaker authentication technique is referred to as verbal information verification [7] which is an application of utterance verification to speaker verification.

A two-pass utterance verification system is shown in Figure 3. Note it is similar to Figure 2 except that the claim identity is now a linguistic unit such as a word or a phrase. The first pass in Figure 3, known as *keyword spotting*, involves the detection of keywords embedded in continuous speech. The second pass, on the other hand, takes the detected keywords, breaks them into subword speech segments, computes corresponding subword verification scores, merges the subword scores into some word and string level *confidence measures* and decides if the "claimed" keywords or key-phrases should be accepted or rejected. We now describe each of the components.

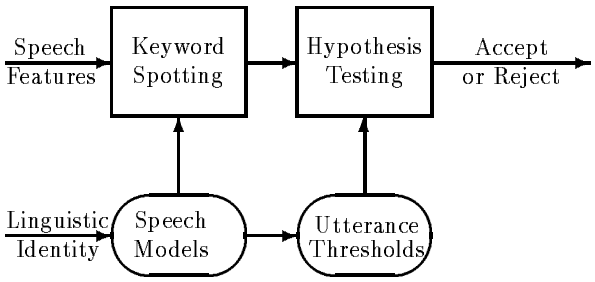

Figure 3: A two-pass utterance verification system

4.1. Keyword and Key-Phrase Spotting

Keyword spotting (e.g. [12, 19]) is the process of detecting a pre-fixed set of words in continuous speech. It often involves both locating and identifying the keywords. In addition to modeling keywords, it is also crucial to have an accurate characterization of non-keyword events. These events, ofter referred to as *fillers*, represent the background in which keywords are extracted. Filler can be as simple as a mixture Gaussian speech model [11], a general HMM speech model, also known as a sink or *garbage* [19] model, a network of context-independent phone models [12], a set of common words and carrier phrases directly preceding and following the keywords, or a combination of the above. Some have even proposed the use of a large vocabulary speech recognition system to perform keyword spotting. In this case, the number of fillers can be several order more than the number of keywords. Despite the heavy computation load some success has been observed. However, it is well known that a good language model is often needed to achieve a reasonable performance of large vocabulary recognition. Such a task-dependent language is usually not available for newly-defined tasks.

4.2. Subword Verification

Subword verification is accomplished in a similar way like speaker verification shown in Eq. (7). The only difference is to replace speaker and anti-speaker models with subword and *anti-subword* models. The speech models that are required to verify the linguistic units are usually composed with a set of phone models and *anti-phone* models. Similar to speaker verification discussed in Section 3, subword verification is more effective when a normalized test statistic, such as the generalized likelihood ratio shown in Eq. (7), is used (e.g. [16]). All the verification strategies we introduced in Section 3.1 and the techniques we presented in Section 3.3 for anti-speaker speech unit modeling can be directly used, modified or extended to

anti-subword modeling. It is also instructive to apply the concept of *phone cohort set*, C_p, which is a set of phone units that are most "close" to the target phone unit, p. In addition to generating cohort based on acoustic distance between speech unit models, one can also construct the phone cohort set based on linguistic closedness (e.g. [16]). So far, we have found that acoustically defined cohort performs better than linguistically defined cohort [16]. Furthermore, it was also found [16] that phone-specific cohort works better than phone-independent cohort. This is similar to our earlier discussion for speaker verification.

It is important to know that verification tests trained with task-specific data outperform those trained with task independent strategies. It is also noted that not all techniques can be applied directly to *vocabulary independent utterance verification* [16]. Even thresholds have to be determined without knowing the task constraints and the vocabularies in such a scenario.

4.3. From Subword to String Verification

When verifying a speech segment, we often perform subword verification first on small segments containing the corresponding subwords and then combine the subword verification scores to form string or word level verification scores, sometimes referred to as a *confidence measure*.

Given a set of subword verification scores with each score, $s_q(X_q, \Lambda)$, corresponding to the q-th speech segment X_q, one way to compose a string score based on the N subword segments in the test utterance X is to again take a geometric average of the subword scores as follows,

$$S(X, \Lambda) = \ln[\frac{1}{N}\sum_{q=1}^{N} w_q \cdot \exp\{\eta \cdot s_q(X_q, \Lambda)\}]^{\frac{1}{\eta}}, \qquad (9)$$

where w_q is a sub-word weight. Other confidence measures (e.g. [7, 16, 10]) have also beeb implemented. MCE and MVE based training techniques can also be used to optimize the parameters needed to compute these confidence measures.

5. VERBAL INFORMATION VERIFICATION

One interesting application of utterance verification is to verify the content of spoken utterances against the personal information registered in a text-based profile of an individual. This new speaker authentication scenario is known as verbal information verification [7]. Human verification of such information is a routine business practice. In cases where a high transaction security is needed when a user tries to access his or her account remotely, the user is often requested by the system "gatekeeper" to provide verbal answers to some questions that are private to the user such as: "what is your mother's maiden name?", "what is your birthday and your birth place?" or "what is your telephone number and/or your zip code?". In contrast to providing this static information which was recorded in the personal profile at the time of user registration, sometimes the information can also be *dynamic* such as: "when was your last transaction?" or "what was the amount of your last transaction?". VIV is a new technique that is capable of providing the above flexible features to help enhance speaker authentication. A block diagram of a VIV system that handles multiple pass phrases is shown in Figure 4.

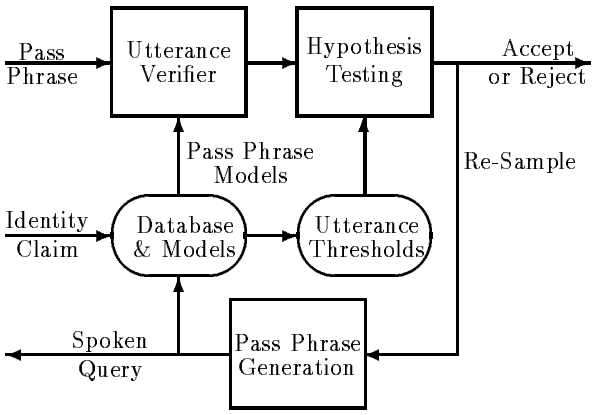

Figure 4: Multi-phrase verbal information verification

The VIV system first accepts the user's identity claim just like what normally performed in a typical SV system. The system then pulls out the personal profile of the claimed user from the system database and generates queries one at a time either randomly or in a fixed order. The system also uses the stored set of subword models and the lexicon that corresponds to the expected answer to a particular query and composes *pass phrase models*. These models are then used by the utterance verifier to compare with the user's pass phrase answer to the system's query. A *confidence measure* of the match between the query and the corresponding answer is now computed. Depending on the goodness of the match with respect to the stored utterance thresholds, the system decides to accept or reject the user's claim indentity or the system may also proceed to *re-sample* the user by generating subsequent queries. This idea is consistent with the idea behind sequential hypothesis testing. In principle, the more the system samples the user, the better the access security of the system. Of course, the service provider could risk losing the user's account if it is too difficult for legitimate users to access their account. A proper sequential verification strategy to compromise the tradeoff needs to be carefully designed.

A possible strategy is to uniformly require the users to conduct a fixed number of queries with the system. Only the users who successfully pass all the queries will be granted system access. In other words, the user is denied access as soon as he or she fails the first test in the sequence of queries. In an experimental VIV system [7], such a strategy was found to achieve perfect verification performance in a series of only three questions. Other scenarios can also be designed depending on various system requirements. This new VIV capability allows us to design novel verification strategies to meet the increasing challenge from new applications that demand for better system performance and enhanced system security.

In addition to the enhanced capabilities, the VIV strategy also offers additional advantages. Most of all, the pass phrase models can be composed with a set of speaker independent models. Therefore no speaker enrollment is required. This alleviates the difficulty in acquiring speaker-dependent models reliably and securely. There is also no

need to manage and maintain a large set of speaker models and their corresponding thresholds for a large population of users. However caution is needed for the user to maintain the privacy of the information in his or her personal profile because other imposters can also access the user's account without the need of imitating the user's voice characteristics. If this is a major concern in the system design, one can always combine SV and VIV strategies to take advantage of the features in both systems.

6. DISCRIMINATIVE TRAINING FOR HYPOTHESIS TESTING

We have discussed how likelihood ratio tests can be designed to perform speaker and utterance verification. It is obvious that in practice, we don't have the exact knowledge about the distributions of H_0 and H_1, which are usually characterized by HMMs. Therefore the optimal properties of the likelihood ratio tests can no longer be guaranteed. The corresponding HMM parameters need to be estimated from the set of available training data usually by ML. estimation (e.g. [9]). One way we can do better is to borrow from the recent success of applying the MCE and MVE discriminative training framework (e.g. [4, 8]) to statistical speech and speaker recognition and extend it to the statistical hypothesis testing problems. This new paradigm, aiming at achieving minimum verification error, not only improves our understanding on the use of likelihood ratio tests in the HMM-based testing framework but also provides a theoretical basis for designing "optimal" tests that minimizes the empirical errors of the training set.

The MCE and MVE formulations are quite similar. They both fall into the general area of *decision-feedback learning*. In the following we present the MCE/MVE training strategy, highlight the difference from conventional ML training paradigm and discuss the family of *generalized probabilistic descent* (GPD) algorithms for solving the parameter estimation problems in MCE/MVE training.

6.1. From ML to MCE/MVE Training

The ML training algorithms assume the form of the HMM distribution and estimate the unknown HMM parameters for a given speaker model using only the data provided by the same speaker. However, such a *distribution estimation* approach does not guarantee an optimal performance for either speaker or utterance verification for the following three reasons : (1) Incorrect model assumption: the assumption about a speech model being an HMM is not entirely true; (2) Insufficient training data : only a small amount of training data is available for training speech models; (3) Inconsistent training and testing : the testing objective aims at minimizing the error rate while the ML training formulation does not necessarily imply a minimum error rate for the training data.

In the last few years, the MCE and the MVE formulation have been proposed to provide an alternative solution to parameter estimation problems in speech recognition [4]. One of the main reasons for the superior performance of the MCE/MVE formulation is that such algorithms solve for the model parameters that minimize the recognition or verification errors of the given set of training data. This is consistent with the goal for designing optimal pattern recognizer and verifyer.

6.2. MCE/MVE Formulation and GPD Algorithm

The GPD training algorithm is based on a smooth embedding of three functions defined as follows:

(a) *Discriminant function* $g_k(X_i;\Lambda)$ for class C_k which is often defined as $g_k(X_i;\Lambda) = \ln[\mathcal{L}_k(X_i)]$. Therefore the discriminant function $g_k(X_i;\Lambda)$ evaluates the log-likelihood of class Q_k upon observing the pattern X_i.

(b) *Misclassification or misverification measure*

$$d_k(X_i;\Lambda) = -g_k(X_i;\Lambda) + G_k(X_i;\Lambda) \qquad (10)$$

where $G_k(X_i;\Lambda)$ is considered as an *anti-discriminant function* of the input pattern X_i in class k. It is a collective representation of all the other competing classes with respect to class k and can be defined as

$$G_k(X_i;\Lambda) = \ln\{\frac{1}{K}\sum_{j\in Q_k}\exp[\eta g_j(X_i;\Lambda)]\}^{\frac{1}{\eta}}, \qquad (11)$$

i.e. the logarithm of the geometric mean of the likelihoods of all K other competing classes. The parameter η can be considered as a weighting factor for adjusting contribution of discriminant functions of other competing speech units. When a large value of η is used, the anti-discriminant function is Eq. (9) is dominated by the most competing discriminant functions, i.e. as $\eta \to \infty$, $G_k(X_i;\Lambda) \to \max_{j\neq k} g_j(X_i;\Lambda)$. The measure $d_k(X_i;\Lambda)$ can be considered as a generalized distance between the combined discriminant function of all the competing classes and the true class for X_i. It can also be considered as a *generalized log likelihood ratio* between the discriminant and the anti-discriminant functions.

(c) *Loss function* $l_k(X_i;\Lambda) = \ell(d_k)$ which is specified as a smooth 0-1, non-decreasing function of d_k to serve as an approximate error count. In this study we use a monotonic Sigmoid loss function of the form

$$\ell(d_k) = \frac{1}{1+\exp[-a(d_k+b)]}, \qquad (12)$$

where a is a positive constant indicating the slope of the Sigmoid function near the decision boundary $d_k + b = 0$.

Minimum error based discriminative training [8] becomes finding the set of parameters Λ that minimize the empirical average cost (or error rate) over all V samples,

$$L(\Lambda) = \frac{1}{V}\sum_{i=1}^{V}\sum_{k=1}^{K} l_k(X_i;\Lambda)1(X_i \in Q_k) \qquad (13)$$

where $1(\mathcal{S})$ is the indicator function for a logical variable $\mathcal{S}$ defined as

$$1(\mathcal{S}) = \begin{cases} 1 & \text{if } \mathcal{S} \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \qquad (14)$$

In pattern verification for each unit to be verified, we can separate the training samples into two types, the correct and the impostor classes, and define the empirical type I and type II errors as follows,

$$L_1(X,\Lambda) = \frac{1}{V_1}\sum_{i=1}^{V_1}\sum_{k=1}^{K} l_k(X_i,\Lambda)\cdot 1(X_i \in Q_k) \qquad (15)$$

and

$$L_2(X,\Lambda) = \frac{1}{V_2}\sum_{i=1}^{V_2}\sum_{k=1}^{K} l_k(X_i,\Lambda)\cdot 1(X_i \in Q_{\bar{k}}) \qquad (16)$$

where V_1 and V_2 are the numbers of samples in the two classes. The weighted total error $L(X,\Lambda) = \omega_1 L_1(X,\Lambda) + \omega_2 L_2(X,\Lambda)$ can now be optimized. The weights are determined depending on the application requirements.

Λ is adjusted from one iteration to the next by

$$\Lambda_{t+1} = \Lambda_t + \delta\Lambda_t \tag{17}$$

with Λ_t being the parameter set at the $t-$th iteration. The *correction* term $\delta\Lambda_t$, in a *batch* mode, is defined as

$$\delta\Lambda_t = -\epsilon_t V_t \nabla L(\Lambda_t) \tag{18}$$

where V_t is a positive definite *learning matrix* and the *learning step size* ϵ_t is a small positive real number. In such cases, the GPD adjustment is performed only when all the training data have been processed. This procedure is therefore referred to the *epoch* training mode. The detailed estimation procedure for HMM parameters by the segmental GPD algorithm can be found in [4, 8].

Alternatively, $\delta\Lambda_t$ can be solved sequentially every time a training sample X_t from class Q_k is given, i.e.

$$\delta\Lambda_t = -\epsilon_t V_t \nabla l_k(X_t;\Lambda_t). \tag{19}$$

It can be shown that the above GPD adjustment rule results in an estimate Λ_t that converges to a solution $\hat{\Lambda}$ that locally minimizes the empirical error rate in a probabilistic sense [4, 10].

7. SUMMARY

Due to the increasing demands in remote access of personal information and transactions over public and private networks, *speaker and speech authentication* based on a user's voice is becoming a natural way of enhancing security because the telephone is still the most prevailing means of communication in business and households worldwide. The newly emerging usage of mobile communication devices and the internet also poses new challenges for the authentication technology.

We have presented a unified statistical hypothesis testing approach to speaker and speech verification. Other biometric pattern verification problems, such as verification of signature, face, palm, fingerprint, gesture, dynamic medical registration such as ECG and EEG, and combination of these signals, can also benefit from the formulations discussed in this paper.

REFERENCES

[1] D. K. Burton, "Text Independent Speaker Verification Using Vector Quantization Source Coding," *IEEE Tran. Acoust., Speech, Signal Processing*, Vol. ASSP-35, Feb. 1987.

[2] A. Higgins, L. Bahler and J. Porter, "Speaker Verification Using Randomized Phrase Prompting," *Digital Signal Processing* Vol. 1, pp. 89-106, 1991.

[3] A. Higgins and L. Bahler, "Text-Independent Speaker Verification by Discriminator Counting," *Proc. ICASSP-91*, Toronto, Vol. 1, pp. 405-408, May 1991.

[4] B.-H. Juang, W. Chou and C.-H. Lee, "Discriminative Methods for Speech Recognition," *IEEE Trans. on Speech and Audio Processing*, Vol. 5, No. 3, pp. 257-265, May 1997.

[5] C.-H. Lee, "Fundamentals of Speaker and Utterance Verification," *ICASSP-97 Tutorial Notes*, Munich, April 1997.

[6] E. L. Lehmann, *Testing Statistical Hypotheses*, Wiley, New York, 1959.

[7] Q. Li, B.-H. Juang, Q. Zhou and C.-H. Lee, "Verbal Information Verification," *Proc. EuroSpeech-97*, Greece, September 1997.

[8] C.-S. Liu, C.-H. Lee, W. Chou, A. E. Rosenberg and B.-H. Juang, "A Study on Minimum Error Discriminative Training for Speaker Recognition," *Jour. Acoust. Soc. Am.*, Vol. 97, No. 1, pp. 637-648, Jan. 1995.

[9] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, Vol. 77, pp. 257-286, 1989.

[10] M. Rahim and C.-H. Lee, "String-Based Minimum Verification Error (SB-MVE) Training for Speech Recognition," *Computer, Speech and Language*, Vol. 11, No. 2, pp. 147-160, April, 1997.

[11] D. A. Reynolds and R. C. Rose, "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models," *IEEE Trans. on Speech and Audio Processing*, Vol. 3, No. 1, pp. 72-83, Jan. 95.

[12] R. C. Rose, "Keyword Detection in Conversational Speech Utterances Using Hidden Markov Model Based Continuous Speech Recognition" *Computer, Speech and Language*, Vol. 9, No. 9, pp. 309-333, 1995.

[13] A. E. Rosenberg, C.-H. Lee, F. K. Soong, and M. A. McGee, "Experiments in Automatic Talker Verification Using Sub-Word Unit Hidden Markov Models," *Proc. ICSLP-90,* Kobe, Japan, Nov. 1990.

[14] A. E. Rosenberg, J. Delong, C.-H. Lee, B.-H. Juang and F. K. Soong, "The Use of Cohort Normalized Scores for Speaker Recognition", *Proc. ICSLP-92,* Banff, pp. 599-602, Oct. 1992.

[15] F. K. Soong, A. E. Rosenberg, L. R. Rabiner and B. H. Juang, "A Vector Quantization Approach to Speaker Recognition," *AT&T Technical Journal,* Vol. 66, pp. 14-26, Mar/Apr 1987.

[16] R. A. Sukkar and C.-H. Lee, "Vocabulary Independent Discriminative Utterance Verification for Non-Keyword in Subword Based Speech Recognition," *IEEE Trans. on Audio and Speech Processing*, Vol. 4, No. 6, pp. 420-429, Nov. 1996.

[17] R. Sukkar and J. G. Wilpon, "A Two Pass Classifier for Utterance Rejection in Keyword Spotting," *Proc. ICASSP-93*, Vol. II, pp. 451-454, Minneapolis, 1993.

[18] N. Tishby, "On the Application of Mixture AR Hidden Markov Models to Text Independent Speaker Recognition," *IEEE Tran. Acoust., Speech, Signal Processing*, Vol. ASSP-39, pp. 563-570, March 1991.

[19] J. G. Wilpon, L. R. Rabiner, C.-H. Lee and E. R. Goldman, "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models," *IEEE Trans. Acoustic, Speech and Signal Proc.*, Vol. ASSP-38,, No. 11, pp. 1870-1878, 1990.