AUTOMATIC ENROLLMENT FOR SPEAKER AUTHENTICATION

Qi Li, Hui Jiang, Qiru Zhou, Jinsong Zheng

Multimedia Communications Research Laboratory
Bell Labs, Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ 07974, USA
{qli,hui,qzhou,jszheng}@research.bell-labs.com

ABSTRACT

Enrollment is a necessary session in speaker verification. Automatic verification of collected training utterances, so called automatic enrollment, is critical to the performance of any speaker verification system. In this paper, we propose one solution including two separate approaches for automatic enrollment. First, at the utterance level, we propose to use an N-best algorithm to perform spoken content verification for training utterance selection. Second, at the word level, we employ an accurate utterance verification algorithm to conduct word verification for training data selection. Finally, we combine the two techniques together and propose a solution for automatic enrollment. Our experiments show that the N-best approach and the utterance verification approach can provide very low error rates. Based on the experimental results, we expect that the combined solution is close to error free and is feasible for real-world applications.

1. INTRODUCTION

As is well known, a typical speaker verification system has two kinds of sessions, enrollment and test. In an enrollment session, an identity, such as an account number, is assigned to the 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 a speaker dependent (SD) hidden Markov model (HMM) is built based on the enrolled utterances in the session. In a test session, the speaker's test utterance is compared against the pre-trained, SD HMM. A speaker is accepted if the matching score exceeds a preset threshold; otherwise the speaker is rejected. For such an SV system, the accuracy of collected training data is critical to the entire system performance. If a wrong utterance is involved in training, we will encounter two kinds of problems. First, the speaker dependent model constructed by the wrong utterance will give a poor performance, and second, once the model is build, it is difficult to correct the error in real applications. Unfortunately, for human, this kind of errors is unavoidable. A speaker may make a mistake while repeating the training utterances.

In [3], we proposed to use verbal information verification (VIV) [1, 2] to collect and to verify training data. Using the method, a speaker is first verified by VIV. After the speaker accesses the account for a few times, usually 4 to 5 times, an SD HMM is trained using the recorded pass-phrases of previous accesses. Then, the authentication process can be switched from VIV to SV. Although the VIV approach is attractive, for the real applications of the traditional speaker verification technology, we still have to address some practical problems.

We now face two challenges from real applications: First, for an on-line enrollment procedure, can we make decision on one, just uttered pass-phrase? If the collected training samples are good, we use them for training; otherwise, we can prompt the user to repeat immediately. Second, when a user makes a mistake, e.g. one digit is wrong in a 10-digit long utterance, can we detect the wrong digit and use rest of the utterance for training? Similarly, when a user has a strong accent, can we use only part of the training utterance for training (and even for future speaker verification) instead of rejecting the user?

To address these new challenges, we propose one solution including two approaches. First, we employed the N-best search algorithm for utterance-level decisions. Second, we applied an accurate, utterance verification algorithm for word-level decision and word selection. Finally, we combine the above approaches to propose a solution for real-world applications.

2. DATABASE AND MODEL

The experimental database for utterance verification consists of fixed-length telephone numbers. It includes 518 utterances and each utterance has 10 digits recorded over the long distance telephone network. The feature vector is composed of 12 cepstrum and short-term energy, plus their first and second order derivatives, i.e. delta and delta-delta cepstrum coefficients. In total, each feature vector has 39 dimensions. The cepstrum is derived from a 10th order LPC analysis over a 30 ms window. The feature vectors are updated at 10 ms intervals, and the same feature will be used for both utterance verification and speaker verification.

The speaker-independent model for speech recognition and segmentation is a whole-digit and context-dependent model, called head-body-tail (HBT) model [4, 5]. The HBT model assumes that context dependent digit models can be built by concatenating a left-context dependent unit (head) with a context independent unit (body) followed by a right-context dependent unit (tail). In detail, each digit consists of 1 body, 12 heads and 12 tails (representing all left/right contexts), for a total of 276 units, 11(digits) × [1(body) + 12(head) + 12(tail)] + 1(silence) [4]. We typically use a 3-state HMM to represent each head and tail unit, and a 4-state HMM for each body. Overall it corresponds to a 10-state digit model for a total number of 837 states (including a 1-state silence model). The model was trained using several large telephone databases (excluding the above one) recorded in the landline and wireless telephone networks. The model was trained initially by maximum likelihood estimation and then modified by the MCE/GPD algorithm [4, 6] to further improve the robustness. When using the model for speech recognition on the above database, it can obtain a string error rate of 6.95%.

3. N-BEST APPROACH FOR TRAINING UTTERANCE SELECTION

The traditional Viterbi search for speech recognition is to find the best path, which has the highest likelihood score. In real applications, since the pronunciations of some words are highly confusable, the correct word sequence may not be in the best path, but in the 2nd or the 3rd path. To address the problem, N-best decoding algorithms were developed to provide multiple utterance hypotheses, then other knowledge sources, error protection measures, or hypothesis testing can be incorporated to further improve the recognition accuracy.

The N-best algorithm used in this paper is based on the tree-trellis algorithm originally proposed by Soong and Huang [7], and later extended by Wu, *et al.* in [8]. The search consists of two parts: a forward, time-synchronous, trellis search and a backward, time asynchronous, tree search. In the forward search, the well-known Viterbi algorithm is used for finding the best hypothesis and for preparing a map of all partial paths scores in a time synchronous scheme. In the backward search, an A^* kind of search is conducted to grow partial paths backward in a time asynchronous scheme. Each partial path in the backward tree search is rank ordered in a stack by the corresponding full path score, which is computed by adding the partial path score with the best possible score of the remaining path obtained from the trellis path map. In each path growing cycle, the current best partial path, which is at the top of the stack is extended by one word [7]. The algorithm also allows the incorporation of inter-word context dependent models and language models in both forward and backward search directions [8].

A block diagram of the N-best approach for automatic enrollment is shown in Fig. 2. After the user claims the identity, the user's pass-phrase, P, in text is retrieved from a database. When the pass-phrase is uttered, the N-best algorithm is employed to search through the utterance and find "N" candidates of recognized word sequences, $\{S_i\}_{i=1}^N$. Each of the candidates is then compared with the retrieved pass-phrase. If P is included in the "N" candidates, the uttered pass-phrase is accepted for training, otherwise it is rejected:

$$\begin{cases} \text{Acceptance:} & P = S_j \in \{S_i\}_{i=1}^N; \\ \text{Rejection:} & \text{otherwise}. \end{cases} \tag{1}$$

If a rejected pass-phrase is a correctly uttered one, the error is called false rejection (FR). On the other hand, if an accepted pass-phrase is an incorrectly uttered one, the error is called false acceptance (FA).

To evaluate the FR rate, we used the database and the HBT model introduced in section 2 to conduct an experiment using the N-best search algorithm. The results are listed in Table 1. When only used the regular one-best search, the FR rate was 6.95%. When used the 2-best search, in addition to the 1-best search, the FR rate was reduced by 2.9%. Furthermore, when used the 3-best search, the FR rate was reduced by another 1.15%. In total, the overall FR rate was 2.90%, where the N-best search algorithm recovered 4.05% correctly uttered pass-phrases. A summary of the final result is shown in Table 2.

Table 1: **Comparison on the N-Best Approaches for Training Utterance Selection**

Descriptions	String False Rejection (%)
1-Best (Traditional ASR)	6.95%
+ 2-Best	-2.90%
+ 3-Best	-1.15%
Total (3-Best)	2.90%

Table 2: **Summary on the N-Best Approach for Training Utterance Selection**

Utterance False Acceptance	Utterance False Rejection
0.0%	2.9%

4. UTTERANCE VERIFICATION FOR TRAINING DATA SELECTION

In this section, we investigate the utterance verification (UV) approach to automatically verify the quality of each pronounced word in training utterances during the on-line enrollment. If the word is acceptable, we use the corresponding data to train speaker-dependent models; otherwise, reject it and use the remaining correct words for training.

Like other utterance verification algorithms (e.g. [9]), we first use the HBT model to partition utterance **O** into a sequence of segments $\mathbf{O} = \{O_i\}_{i=1}^{K}$ through forced alignment since the transcription is given. Each of the segments corresponds to a subword. For the HBT model, a subword is a "head", "body" or "tail" unit. We then use a pair of context-independent, positive and negative models, λ_i and $\bar{\lambda}_i$ trained for each subword, to compute the log likelihood ratio (LLR). Then, we combine the subword LLRs to a word score for decision. For example, for digit j, we compute an average LLR score as the following confidence measure [10]:

$$M_j = \frac{1}{T}\sum_{i}^{i+2} \ln \frac{P(O_i|\lambda_i)}{P(O_i|\lambda_i)} \qquad (2)$$

where T is the total number of frames corresponding to digit j. Segment O_i, O_{i+1}, and O_{i+2} correspond to the "head", "body" and "tail" unit of digit j, respectively. A decision can then be made on digit j as:

$$\begin{cases} \text{Acceptance:} & M_j > \tau; \\ \text{Rejection:} & \text{otherwise.} \end{cases} \qquad (3)$$

where τ is a threshold value.

In addition to the HBT models which we have used for segmentation, for each subword unit i, we trained a speaker independent, positive model λ_i and a negative model $\bar{\lambda}_i$. The data selection strategy for training the models is adapted from [10]. Basically, we decode every training utterance with the Viterbi beam search algorithm. During the search, we examine all hypothesized subword segments in all word-ending partial paths within the beam. Then each of the hypothesized subword segments is compared with the reference segmentation to decide whether it is a true or a competing token of the subword unit. After collecting the tokens from the entire training database, all true tokens for subword i are used to estimate the positive model λ_i while all competing tokens for subword i are used to estimate the negative model $\bar{\lambda}_i$. The detail procedure for training token selection is in [10].

The above utterance verification approach was evaluated using the database introduced in section 2. We created two cases: A and B, to simulate substitution and deletion errors, respectively. In case A, for every 10-digit long transcription, we randomly change one digit at a random location to simulate the case where the user mispronounces one digit when uttering the pass-phrase. In case B, for every 10-digit long transcription, we randomly insert one digit to a random location to simulate the case where the user misses one digit when uttering the pass-phrase. The verification error rates were plotted as two ROC curves in Fig. 1. From the curves we can find that the word-level EERs (equal error rates) for case A and B are 3.7% and 5.6%, respectively. The reason that the case B is worse is due to the higher segmentation error from forced alignment when the pronounced number of digits differs from the transcription. The results indicate that even when a user makes a mistake; we still can collect training data with very low error rates. We do not have to reject entire utterance - only the mispronounced words.

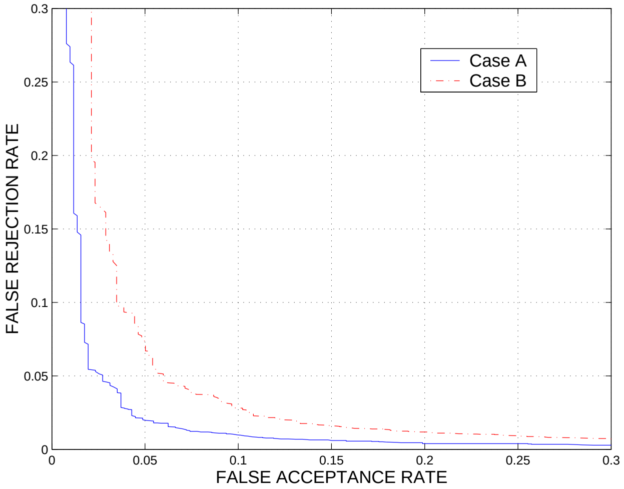

Figure 1: ROC curves of verification for case A (substitution error) and case B (deletion error), at word level.

5. SOLUTION FOR REAL-WORLD APPLICATIONS

So far, we have introduced two approaches for automatic enrollment. Although both of the techniques already have small errors, a combined solution can further reduce the error at the system level and can achieve a close to error-free solution to meet the requirement of real-world applications. The idea is to use the N-best approach to do utterance-level verification, followed by the utterance verification approach to do word-level data selection. The concept is shown in Fig. 2. In the figure, after the N-best search on all the training utterance, a decision is made on the number of rejected

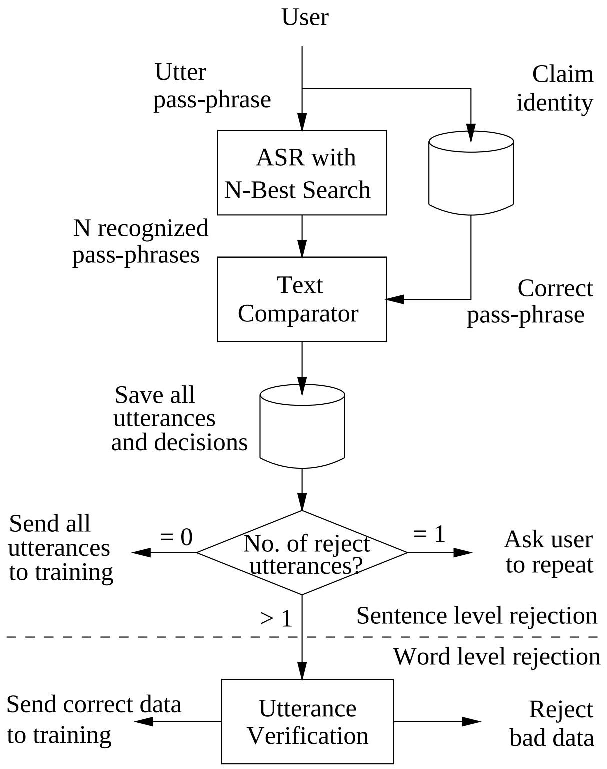

Figure 2: Proposed solution for automatic enrollment.

utterances, R:

$$\begin{cases} R = 0, & \text{All the utterances are correct.} \\ & \text{Send all utterances to training;} \\ R = 1, & \text{Only one utterance is mispronounced.} \\ & \text{Ask the user to repeat;} \\ R > 1, & \text{More than one utterance were mispronounced.} \\ & \text{Send utterances to word-level verification.} \end{cases} \tag{4}$$

For example, a verification system collected 5 training utterances from enrollment. If the proposed N-best approach accepted all the utterances except one, it means that the speaker correctly uttered the pass-phrase 4 times and mispronounced once. The system can then prompt the speaker to repeat. This approach should handle most of the error cases. In some special case, due to a strong accent, the N-best approach might reject all 5 or most of the utterances. Then, the utterance verification algorithm can be applied to examine the utterance digit by digit. Here, we can consider the sequence of the LLR scores computed from an utterance as a pattern. If all the patterns match or are consistent from utterance to utterance, the utterances can still be accepted. However, the digit with lower LLR scores should be excluded from training.

6. CONCLUSIONS

In this paper, we investigated two approaches for automatic enrollment at the utterance and the word levels, and proposed a combined solution for real-world applications. We expect that such a solution be close to error free and feasible to real-world applications. Although we focused our discussions on speaker verification (SV), it is straightforward to apply the techniques to verbal information verification (VIV), such as more accurate spoken content verification and other VIV applications.

7. ACKNOWLEDGMENT

The authors wish to thank F. K. Soong, O. Siohan, A. C. Surendran, and J. P. Olive for useful discussions.

8. REFERENCES

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

[2] Q. Li, B.-H. Juang, Q. Zhou, and C.-H. Lee, “Automatic verbal information verification for user authentication,” *IEEE Trans. on Speech and Audio Processing*, vol. 8, pp. 585–596, Sept. 2000.

[3] Q. Li and B.-H. Juang, “Speaker verification using verbal information verification for automatic enrollment,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Seattle), May 1998.

[4] W. Chou, C.-H. Lee, and B.-H. Juang, “Minimum error rate training of inter-word context dependent acoustic model units in speech recognition,” in *Proceedings of Int. Conf. on Spoken Language Processing*, pp. 432–439, 1994.

[5] C.-H. Lee, E. Giachin, L. R. Rabiner, R. Pieraccini, and A. E. Rosenberg, “Improved acoustic modeling for large vocabulary speech recognition,” *Computer Speech and Language*, vol. 6, pp. 103 – 127, 1992.

[6] B.-H. Juang, W. Chou, and C.-H. Lee, “Minimum classification error rate methods for speech recognition,” *IEEE Trans. on Speech and Audio Process.*, vol. 5, pp. 257–265, May 1997.

[7] F. K. Soong and E. F. Huang, “A fast tree-trellis search for finding the N-best search hypotheses in continuous speech recognition,” *J. Acoust. Soc. AM., S-1*, vol. 87, pp. 105–106, May 1990.

[8] W. Chou, T. Matsuoko, B.-H. Juang, and C.-H. Lee, “An algorithm of high resolution and efficient multiple string hypothesization for continuous speech recognition using inter-word models,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 153–156, 1994.

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

[10] H. Jiang, F.K. Soong, and C.-H. Lee, “A data selection strategy for utterance verification in continuous speech recognition,” in *Proceedings of European Conference on Speech Communication and Technology*, (Aalborg Denmark), pp. pp. 2573–2576, Sept. 2001.