IMPROVED PHONETIC SPEAKER RECOGNITION USING LATTICE DECODING

Andrew O. Hatch[1,2]*, Barbara Peskin*[1]*, and Andreas Stolcke*[1,3]

[1]The International Computer Science Institute, Berkeley, CA, USA
[2]The University of California at Berkeley, USA
[3]SRI International, Menlo Park, CA, USA
{ahatch,barbara}@icsi.berkeley.edu, stolcke@speech.sri.com

ABSTRACT

The current "state-of-the-art" in phonetic speaker recognition uses relative frequencies of phone n-grams as features for training speaker models and for scoring test-target pairs. Typically, these relative frequencies are computed from a simple 1-best phone decoding of the input speech. In this paper, we present results on the Switchboard-2 corpus, where we compare 1-best phone decodings versus lattice phone decodings for the purposes of performing phonetic speaker recognition. The phone decodings are used to compute relative frequencies of phone bigrams, which are then used as inputs for two standard phonetic speaker recognition systems: a system based on log-likelihood ratios (LLRs) [1, 2], and a system based on support vector machines (SVMs) [3]. In each experiment, the lattice phone decodings achieve relative reductions in equal-error rate (EER) of between 31% and 66% below the EERs of the 1-best phone decodings. Our best phonetic system achieves an EER of 2.0% on 8-conversation training and 1.4% when combined with a GMM-based system.

1. INTRODUCTION

Most conventional speaker recognition systems use Gaussian mixture models (GMMs) to capture frame-level characteristics of a person's voice, where the speech frames are assumed to be independent of one another. Because of this independence assumption, GMMs often fail to capture certain types of speaker-specific information that evolve over time scales of more than 1 frame. For example, since words usually span many frames, GMMs tend to be poorly suited for modeling differences in word usage (idiolect) between speakers. In [4], Doddington used word n-grams to model speaker-specific patterns of word usage.

Another recent effort at moving beyond the standard GMM-based paradigm is to explicitly model phone sequences used by speakers. This line of research, which is generally referred to as *phonetic speaker recognition*, was pioneered by Andrews et al., who used relative frequencies of phone n-grams to capture sequential patterns in an individual's speech [1, 2]. This work was subsequently extended in various papers, such as the work of the "SuperSID" team at the JHU 2002 Summer Workshop [5, 6, 7]. In 2003, Campbell et al. used support vector machines (SVMs) to train phonetic speaker models [3].

While these phonetic approaches have generally been quite effective, it is our opinion that the true potential of phonetic speaker recognition has yet to be realized—mainly because past systems have used 1-best decodings instead of lattice decodings to estimate relative frequencies of phone n-grams. In this paper, we compare 1-best phone decodings vs. lattice phone decodings for the purposes of performing phonetic speaker recognition. The phone decodings are used to compute relative frequencies of phone bigrams, which are then used as inputs for two standard phonetic speaker recognition systems: a conventional system based on log-likelihood ratios (LLRs) [1, 2], and an SVM-based system similar to that of Campbell et al. [3]. The results indicate that lattice decodings provide a much richer sampling of phonetic patterns than 1-best decodings. Note that a similar comparison between decoding methods—with similar results—was recently reported within the field of language recognition [8]. We were unaware of this work until the time of its publication, shortly after the submission of this paper.

The paper is organized as follows: Section 2 describes the task and dataset. Section 3 describes our phone recognition system and outlines a procedure for estimating relative frequencies of phone bigrams. Section 4 describes the metrics that were used to train speaker models and score test-target pairs. Section 5 describes the experiments that we performed and discusses the results. Finally, section 6 provides a summary of our findings.

2. TASK AND DATA

The experiments reported in this paper were performed on the NIST 2003 Extended Data task, which uses phases II and III of the Switchboard-2 corpus. The combination of phases II and III amounts to a total of 14257 conversation sides with an average length of approximately 2.5 minutes of speech. Under the Extended Data task, the conversation sides are divided into 10 splits. There are no common speakers between any two splits. Thus, when one split is selected for testing, the remaining 9 splits can be used to train a background model.

The NIST Extended Data task comprises 5 different training conditions: 1, 2, 4, 8, and 16-conversation training. In this paper, we provide results for 1-conversation and 8-conversation training. These two training conditions provide a reasonably informative view of how speaker recognition accuracy is affected by the amount of available training data.

3. PHONETIC PROCESSING

The following section describes the steps involved in computing relative frequencies of phone bigrams from conversation sides.

This material is based upon work supported by the National Science Foundation under grant No. 0329258

3.1. Speech/non-speech detection

For the experiments in this paper, we used a speech/non-speech detector developed at SRI International [9] to remove non-speech frames from the input audio. This step breaks the original conversation sides into smaller chunks containing mostly speech.

3.2. Phone recognition

After removing non-speech frames from the input conversation sides, we used the DECIPHER speech recognition system [9] developed by SRI International to perform both 1-best and lattice phone decoding. Our particular version of DECIPHER uses gender-dependent, monophone acoustic models, where each monophone is modeled by a 3-state hidden Markov model (HMM). The acoustic model was trained on the Switchboard 1 corpus using MFCC features. Note that the phone decodings were performed in open-loop mode (i.e. we used a unigram phone language model with uniform probabilities).

3.3. Estimating relative frequencies of phone bigrams

3.3.1. 1-best phone decoding

Given a phone decoding (either 1-best or lattice), the next step is to compute the relative frequency of each phone bigram. For the case of a 1-best phone decoding, this step is straightforward—we simply count the number of times each phone bigram appears in the hypothesized phone sequence and then divide by the total number of bigrams. Given some input audio, X, we have:

$$p(d_i|X) = \frac{count(d_i|X)}{\sum_{k=1}^{M} count(d_k|X)} \quad (1)$$

Note that $d_1, \ldots, d_M$ represents the set of unique phone bigrams, and $count(d_i|X)$ refers to the number of times phone bigram d_i appears in the decoding of the input audio, X. The term, $p(d_i|X)$, represents the observed *relative frequency* of d_i, which we can interpret as the sample probability of d_i within the 1-best decoding of X.

3.3.2. Lattice phone decoding

For the case of a lattice phone decoding, we can use the following equation to compute the expected count of phone bigram d_i given an input speech signal, X:

$$E[count(d_i|X)] = \sum_{Q} p(Q|X) \cdot count(d_i|Q) \quad (2)$$

Here, Q represents a hypothetical phone sequence corresponding to the entire utterance, X, and $p(Q|X)$ represents its posterior probability, as determined by the phone recognizer. The term, $count(d_i|Q)$, refers to the number of times d_i appears within phone sequence Q. A standard forward-backward approach can be used to efficiently compute the expected counts in equation (2). Once we have a complete set of expected counts, we can use equation (1) to convert them into relative frequencies.

4. MODEL TRAINING AND SCORING

In this section, we describe two methods for training speaker models and for scoring test-target pairs (i.e. a particular target speaker model and test conversation side that are scored against each other).

4.1. The log-likelihood ratio (LLR) approach

The traditional method for scoring a test-target pair in a phonetic speaker recognition system is to compute the log-likelihood ratio (LLR) of the target speaker model vs. a background model. For this paper, we use the following equation to compute the LLR for test conversation side A and target speaker model B:

$$LLR(A,B) = \sum_{i=1}^{M} p(d_i|convSide_A) \log \frac{p(d_i|spk_B)}{p(d_i|bkg)} \quad (3)$$

Here, $p(d_i|convSide_A)$, $p(d_i|spk_B)$, and $p(d_i|bkg)$ refer to the probability of phone bigram d_i for conversation side A, speaker model B, and for the background model, respectively. For the experiments in this paper, we used relative frequencies computed from splits 6 through 10 as a background model—that is, to estimate the $p(d_i|bkg)$ terms in equation (3)—for all speaker models belonging to splits 1 through 5, and vice versa. Similarly, we used relative frequencies derived from the training conversation sides of speaker B to estimate $p(d_i|spk_B)$ for all $i \in 1, \ldots, M$. Note that the form of the LLR used in equation (3) is equivalent to that used in [1], [2], and [3].

Given the large number of speakers that were used to train the background models (each background model comprises over 500 speakers), we might expect our estimates of $p(d_i|bkg)$ to be reasonably robust. The amount of data available for training the speaker models, on the other hand, is considerably less—only 1 or 8 conversation sides, depending on the number of training conversations. Thus, our estimates of $p(d_i|spk_B)$ may be fairly noisy, particularly for the case of 1-conversation training. To make our probability estimates more robust, we applied the following linear smoothing model to relative frequencies extracted for a given speaker:

$$p_s(d_i|spk_A) = (1-\alpha) \cdot p(d_i|spk_A) + \alpha \cdot p(d_i|bkg) \quad (4)$$

In the above equation, $p_s(d_i|spk_A)$ represents the smoothed relative frequency of d_i, which we compute by taking a weighted average of $p(d_i|spk_A)$ and $p(d_i|bkg)$. The parameter, α, determines the amount of smoothing, and can be set anywhere between 0 and 1. Note that a similar smoothing model was independently developed and reported by Baker, et al. in [10].

We can also define an analogous model for smoothing relative frequencies extracted from conversation sides (i.e. the $p(d_i|convSide_A)$ terms in equation (3)). However, in practice, we have found the benefits of smoothing the $p(d_i|convSide_A)$ terms to be fairly negligible, at least for the purposes of computing LLR scores. For this reason, we only apply smoothing to relative frequencies that correspond to speaker models (i.e. the $p(d_i|spk_B)$ terms).

4.2. The support vector machine (SVM) approach

In their 2003 NIPS paper, Campbell et al. demonstrated a phonetic speaker recognition system based on support vector machines [3]. One of the main innovations of the paper was the following "kernelized" version of the log-likelihood ratio:

$$k(A,B) = \sum_{i=1}^{M} \frac{p(d_i|convSide_A)}{\sqrt{p(d_i|bkg)}} \frac{p(d_i|convSide_B)}{\sqrt{p(d_i|bkg)}} \quad (5)$$

The above expression follows from replacing $p(d_i|spk_B)$ with $p(d_i|convSide_B)$ in equation (3) and then applying the approximation, $\log x \approx x - 1$. If we ignore the offset term in the resulting expression, we arrive at the kernel shown in equation (5). Note that $k(A, B)$ is simply an inner product of relative frequencies of phone bigrams, where each relative frequency is divided by the square root of the corresponding relative frequency from the background model. To allow for affine decision boundaries, each vector of relative frequencies can be augmented with a bias term.

Campbell et al. used the kernel in equation (5) to train SVM-based speaker models [3]. For this paper, we used all of the conversation sides in the background model of a given speaker model as negative training examples for that model. This amounts to approximately 6740 negative training examples for every speaker model. To balance the number of positive versus negative examples, we weighted the training errors for each positive example by $\frac{N}{P}$, where P and N represent the total number of positive and negative training examples, respectively.

Given that we apply linear smoothing to relative frequencies when computing LLRs, it might seem reasonable to use a similar form of linear smoothing on the $p(d_i|convSide_A)$ terms in the SVM kernel of equation (5). However, since SVM classifiers are invariant to uniform scaling and shifting of the input feature vectors, applying a linear smoothing model like that of equation (4) would have no effect on the output results of an SVM-based scoring system. For this reason, we do not apply smoothing for any experiments involving SVMs.

5. EXPERIMENTS

Experiments were conducted on the 1-conversation and the 8-conversation training conditions. For each training condition, we ran experiments on all four possible combinations of decoding method and scoring method (i.e. 1-best decoding vs. lattice decoding and LLRs vs. SVMs). As explained in section 4.1, equation (4) was used to smooth the $p(d_i|spk_A)$ terms for the LLR scoring method. Smoothing parameters for equation (4) were trained by finding the value of α that minimizes the equal-error rate (EER) for each combination of decoding method and number of training conversations, as measured on NIST's 2001 Extended Data set. Note that the 2001 Extended Data set uses the Switchboard-1 corpus, which is similar in format to Switchboard-2, but comprises an entirely different set of speakers. To do SVM training and scoring, we used the SVM^{light} package with $c = 1$ [11]. We also included a bias term in the kernel of equation (5).

The EERs for the experiments are listed in table 1, and the corresponding detection-error tradeoff (DET) curves are shown in figure 1. We have also listed the relative reductions in EER that are achieved by using lattice decodings over 1-best decodings in table 1. The results show that the EERs for the lattice decodings are substantially lower than the corresponding EERs for the 1-best decodings. As shown in table 1, the lattice decodings are most successful—both in terms of minimizing EER and in terms of the improvement that they achieve over 1-best decodings—when used in conjunction with SVMs.

To provide some perspective on these results, we note that the EERs in table 1 for "LLRs, 1-best decoding" compare favorably with those reported by Campbell et al. in [3] for the NIST 2003 Extended Data task (Campbell et al. reported EERs of 21.8% and 8.8% for LLR systems using 1 and 8-conversation training, respectively). For the case of SVM-based scoring, Campbell et al.

	# Training Conversations	
	1	8
LLRs, 1-best decoding	16.4%	6.1%
LLRs, lattice decoding	10.5%	4.2%
relative EER reduction using lattice decoding	*36%*	*31%*
SVMs, 1-best decoding	18.2%	5.9%
SVMs, lattice decoding	8.5%	2.0%
relative EER reduction using lattice decoding	*53%*	*66%*

Table 1. EERs and relative reductions in EER from using lattice decodings vs. 1-best decodings

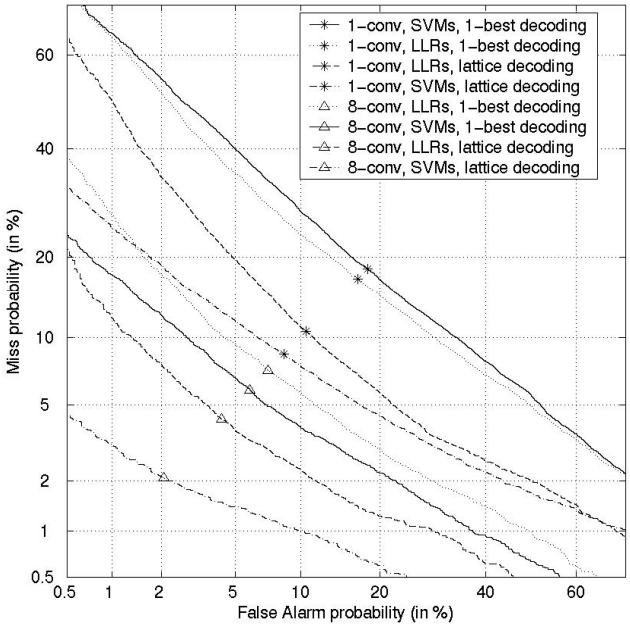

Fig. 1. corresponding DET curves for the systems in table 1

reported EERs of 13.4% and 3.5% [3], which outperform those of our "SVMs, 1-best decoding" systems. However, we note that Campbell et al. used a "phonetic refraction" system, which makes use of multiple phone decodings obtained from recognizers trained on various languages. In spite of the fact that our system only uses a single English phone recognizer, the EERs achieved by our "SVMs, lattice decoding" systems are, to the best of our knowledge, the lowest reported for a phonetic speaker recognition system on the NIST 2003 Extended Data task.

These results suggest that lattice decodings may yield more robust estimates of the relative frequencies of phone bigrams than 1-best decodings. To support this claim, we can examine the α parameters that were used to smooth estimates of $p(d_i|spk_A)$ in the LLR systems. Table 2 shows the α values that were trained for every combination of decoding method and number of training conversations. Note that the α values represent the *optimal* amount of smoothing for minimizing EER on the Switchboard-1 data, and should provide some indication of the robustness of the original phone bigram statistics (i.e. lower α values presumably point to

	# Training Conversations	
	1	8
1-best decoding	0.955	0.670
lattice decoding	0.920	0.040

Table 2. α values trained on the Switchboard-1 corpus

	# Training Conversations	
	1	8
phonetic system (SVMs, lattice decoding)	8.5%	2.0%
GMM system	6.6%	2.6%
GMM + phonetic system	5.0%	1.4%

Table 3. EERs for the individual systems and for the combined GMM + phonetic system

more sufficiently-trained models). As shown in the table, the α values are quite large for each of the systems except for the "lattice decoding, 8-conversations" system, where α is only 0.040. Note that in every case, the α values are smaller for the lattice decodings than for the 1-best decodings. Based on this, we might surmise that relative frequencies obtained from lattice decodings tend to be less noisy than those obtained from 1-best decodings.

As a final experiment, we tried combining the output scores from the "SVMs, lattice decoding" systems with scores obtained from a GMM-based speaker recognition system developed at SRI International [9]. The combination was performed by taking a simple weighted average of output scores. We used the Switchboard-1 corpus to train the combination weights. A comparison between the EERs of the individual and combined systems is provided in table 3, and the corresponding DET curves are shown in figure 2. The results show that the phonetic approach used by the "SVMs, lattice decoding" system is, too a significant extent, complementary to the GMM system. According to table 3, the combined system achieves substantial reductions in EER on both the 1-conversation and the 8-conversation conditions.

6. CONCLUSIONS

In this paper, we compared 1-best phone decodings vs. lattice phone decodings for the purposes of performing phonetic speaker recognition. In each experiment, the lattice decodings achieved relative reductions in EER of between 31% and 66% below the EERs of the 1-best decodings. Our best lattice decoding system achieves an EER of 2.0% on 8-conversation training and 1.4% when combined with a GMM-based system. These results support the view that lattice decodings provide a much richer sampling of phonetic patterns within speech than 1-best decodings.

7. ACKNOWLEDGEMENTS

The authors would like to thank the National Science Foundation for funding this research. We would also like to thank Sachin Kajarekar of SRI International for providing us with output scores for SRI's GMM-based speaker recognition system.

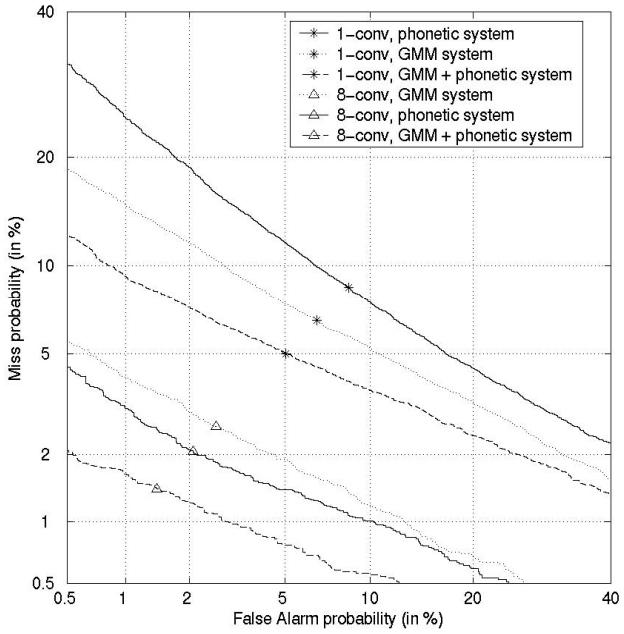

Fig. 2. corresponding DET curves for the systems in table 3

8. REFERENCES

[1] W. D. Andrews, M. A. Kohler, and J. P. Campbell, "Phonetic Speaker Recognition," in *Proc. of Eurospeech*, 2001, pp. 149–153.

[2] W. D. Andrews, M. A. Kohler, J. P. Campbell, J. J. Godfrey, and J. Hernandez-Cordero, "Gender-dependent phonetic refraction for speaker recognition," in *Proc. of ICASSP*, 2002, vol. I, pp. 149–153.

[3] W. M. Campbell, J. P. Campbell, D. A. Reynolds, D. A. Jones, and T. R. Leek, "Phonetic speaker recognition with support vector machines," in *Advances in Neural Information Processing Systems 16*, 2004.

[4] G. Doddington, "Speaker recognition based on idiolectal differences between speakers," in *Proc. of Eurospeech*, 2001, pp. 2521–2524.

[5] Q. Jin, J. Navratil, D. Reynolds, J. Campbell, W. Andrews, and J. Abramson, "Combining cross-stream and time dimensions in phonetic speaker recognition," in *Proc. of ICASSP*, 2003.

[6] J. Navratil, Q. Jin, W. Andrews, and J. Campbell, "Phonetic speaker recognition using maximum likelihood binary decision tree models," in *Proc. of ICASSP*, 2003.

[7] D. Klusacek, J. Navratil, D. A. Reynolds, and J. P. Campbell, "Conditional pronunciation modeling in speaker detection," in *Proc. of ICASSP*, 2003, vol. IV, pp. 804–807.

[8] J. L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Proc. of ICSLP*, 2004.

[9] S. Kajarekar, L. Ferrer, A. Venkataraman, K. Sonmez, E. Shriberg, A. Stolcke, and R. R. Gadde, "Speaker recognition using prosodic and lexical features," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop*, 2003, pp. 19–24.

[10] B. Baker, R. Vogt, M. Mason, and S. Sridharan, "Improved phonetic and lexical speaker recognition through MAP adaptation," in *Proc. of Speaker Odyssey*, 2004.

[11] T. Joachims, "Making large-scale SVM learning practical," in *Advances in kernel methods — support vector learning*, B. Schoelkopf, C. Burges, and A. Smola, Eds. MIT-press, 1999.