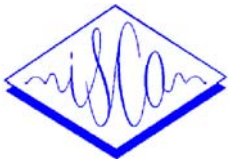

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

ON THE USE OF NEURAL NETWORKS TO COMBINE UTTERANCE AND SPEAKER VERIFICATION SYSTEMS IN A TEXT-DEPENDENT SPEAKER VERIFICATION TASK[1]

L. Rodríguez-Liñares, C. García-Mateo, J. L. Alba-Castro
E.T.S.E. Telecomunicación-Universidade de Vigo
36200 – Vigo (SPAIN)
{leandro,carmen,jalba}@tsc.uvigo.es

ABSTRACT

Speaker Verification and Utterance Verification are examples of techniques that can be used for Speaker Authentication purposes. Speaker Verification consists of accepting or rejecting the claimed identity of a speaker by processing samples of his/her voice. Utterance Verification systems make use of a set of speaker-independent speech models to recognize a certain utterance and decide whether a speaker has uttered it or not. If the utterances consist of passwords, this technique can be used for identity verification purposes. Up to now, both techniques have been used separately. We propose an architecture consisting of both systems working in parallel with a novel output combination technique. Thus, a Neural Network is designed to learn from the data how to balance the influence of both outputs in order to jointly minimize the False Acceptance and False Rejection rates. The better performance of this architecture is compared with those of the individual systems in an over the phone speaker recognition task.

1. INTRODUCTION

Continuous HMM (Hidden Markov Models) based systems are presently the state of the art for speaker recognition purposes [1][2]. They perform a stochastic matching that can be formulated as measuring the likelihood of a collection of vectors given models of the speakers. These vectors are obtained from the voice of the speakers and try to represent the speakers' vocal-tract characteristics during the production of distinct sounds.

Such a Speaker Recognition system does not take into account another important information present as well in the utterance: the message. In prompted-text of password based Speaker Recognition systems, speakers are addressed to pronounce personal utterances that identify them. These utterances are matched against a set of models that represent the vocal tract characteristics of the different sounds regardless of the speaker identity with the purpose of validating the message. Besides, prompted-text systems can be improved by changing the utterances the speakers are addressed to pronounce. This prevents the systems against recordings being used by impostors trying to gain access.

If stochastic matching of the utterances against both speaker models and phone models are performed, we obtain two probabilities: a speaker probability and a message probability. It can be expected that these probabilities are somehow uncorrelated and that the combination of them yield better results than any of them separately. The problem is how to combine the outputs of both sub-systems in order to improve the performance of the final system. In [3] we presented two different methods to perform this combination. In this paper we present a novel method based on a Neural Network structure for combining the Speaker and Utterance Verifiers that improves the overall performance.

The rest of this paper is organized as follows: Section 2 presents the database and Section 3 the Speaker and Utterance Verifiers we have used. In Section 4 the architectures of the dual recognizers are presented. Finally, in Section 5 we present some conclusions and guidelines for further work.

2. EXPERIMENTAL CONDITIONS

The experiments were conducted using our own database, called "TelVoice" [1]. It has been designed for Speaker Recognition purposes and consists of 59 speakers with 10 phone calls each. The time between recordings is variable across speakers, ranging from three days to more than one year.

We have made some choices about recording conditions and speech parametrization. The voice was sampled at 8KHz and off-line filtered to remove the 50 Hz electric-supply noise. Energy and 12 Mel-cepstrum coefficients were computed using a Hamming window with frame length of 25 ms and a frame period of 10 ms. Preemphasis (k=0.97) and liftering (parameter 22) were also used. First and second derivatives of the energy and the Mel-cepstra were appended to the parameters of each frame. This makes a total of 39 parameters per vector.

We conducted the experiments presented in this paper with a subset of this database consisting of 20 speakers (10 males and 10 females) with 5 sessions each one. Each session consists of four repetitions of the Spanish Identity Card number made up of 8 digits. The speakers were addressed to pronounce it naturally (digit by digit, grouping digits or as a whole, as they usually do) but always the same way across sessions.

One of the sessions was used for training models and for calculating thresholds, while the other four sessions were used

[1] This work has been partially supported by Spanish CICYT under projects TIC96-0964-C04-02 and 1FD97-0077-C02-01, and Xunta de Galicia.

for testing: three to simulate clients and one to simulate impostors. That is, the four utterances of the first three sessions where assumed as from "true" speakers (we tested each file against the true identity) and the first utterance of the fourth session were tested against the 20 possible identities of the database. This makes up a total of 260 tests of clients and 380 tests of impostors.

3. INDEPENDENT VERIFIERS

The Speaker Verifier is built up by training a GMM for each speaker with one recording session (approximately 20 seconds) using a Voice Activity Detector (VAD) to identify the noise segments [1]. The GMMs are covariance-tied and the number of gaussian mixtures is 16. In the testing phase, for each verification test we normalized the obtained probability by the probabilities of 6 (3 far and 3 close) speaker's cohorts [2].

The Utterance Verifier is a speaker-independent speech recognizer that makes use of 25 context-independent phone models and a noise model [4][5]. The phone models consist of 3-state left-to-right HMMs with 16 mixtures/state. A forced alignment between the utterance and the chain of models of the expected text is performed using the Viterbi algorithm and the segment likelihoods are normalized using the phone model of the closest competitor as antimodel. The normalized segment probabilities are accumulated and normalized by its lengths.

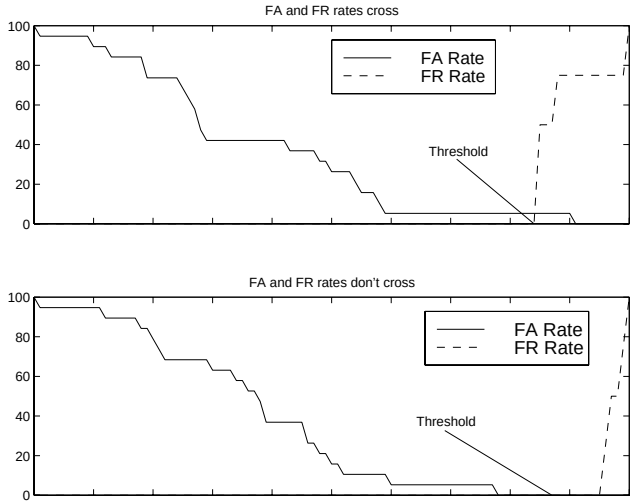

Figure 1. Example of the criteria for taking the values of the thresholds: FA and FR rates obtained with the Utterance Verifier for two speakers of the Database.

We calculate two thresholds per speaker (one per likelihood) just using the training session. For each speaker, the variation of the False Acceptance (FA) and False Rejection (FR) Rates against the threshold were calculated both for the Speaker Verifier and for the Utterance Verifier. In case the False Acceptance Rate and False Rejection Rate cross each other, the threshold corresponding to the point where the False Rejection Rate goes to zero was taken. If the rates don't cross, the used criterion was to take the mean of the thresholds where both rates go to zero. Examples of these cases can be seen in figure 1. The obtained FA and FR Rates for the Speaker and the Utterance Verifiers can be seen in the first two lines of table 1.

4. DUAL VERIFICATION

In this section, we address the problem of how to combine the outputs of both verifiers in order to improve the overall performance.

There are two extreme criteria to accept the claimed identity in the dual tests: to accept the speaker in case one of the probabilities exceeds its threshold (we call it permissive test) or to demand that both probabilities exceed their respective thresholds simultaneously (we call it restrictive test). The results obtained with these two tests correspond to the last two lines of table 1.

Compared with the independent verifiers, the permissive test achieves the lowest FR Rate and the highest FA Rate, while the performance of the restrictive test is just the opposite. The ideal working point should be placed somewhere between these two operating nodes. As a first approach for combining them, we present a structure we called Algebraic Combination (also reported in [2]) that led us to a Neural Network combination. Both architectures are presented in the rest of this section.

Verification	FA Rate	FR Rate
Speaker	3.684%	8.077%
Utterance	12.105%	21.923%
Dual (Permissive)	15.789%	3.846%
Dual (Restrictive)	0.000%	26.154%

Table 1. Results obtained with Independent Verifiers and with Permissive and Restrictive Tests.

4.1 Algebraic Combination

The first approach was to build a continuous function that varies between a permissive and a restrictive test by means of a control parameter α. Let's suppose that L_s and L_u are the speaker and utterance likelihoods, respectively. First, we normalize them by making use of the speaker and utterance thresholds T_s and T_u:

$$\tilde{L}_x = \frac{L_x - T_x}{|T_x|} \quad x = \{s, u\}$$

A sigmoid function was applied for smoothing purposes:

$$S_x = \frac{1}{1 + e^{-a\tilde{L}_x}} \quad x = \{s, u\}$$

Based on previous tests, we took a=5. The scores S_s and S_u take values between 0 and 1 depending on whether they pass the test or they don't.

Making use of these scores we implemented the permissive and restrictive tests (O_{perm} and O_{rest}, respectively):

$$O_{perm} = Sig_2(S_s + S_u - 0.1) - 0.5$$
$$O_{rest} = Sig_2(2S_sS_u - 0.5) - 0.5$$
$$Sig_2(x) = \frac{1}{1 + e^{-bx}}$$

Preliminary experiments led us to a value of b=2. O_{perm} will tend to –0.5 when $L_s<T_s$ and $L_u<T_u$ and to 0.5 if one of the likelihoods is greater than its correspondent threshold. O_{rest} will tend to –0.5 or 0.5 if one of the likelihoods is minor than its

threshold or if both likelihoods are greater than their correspondent threshold, respectively.

At last, the verification test is performed as:

$$V = \alpha\ O_{rest} + (1-\alpha)\ O_{perm}$$
$$V > 0 \quad \Rightarrow \quad Accepted$$

Where α is the control parameter included to balance both behaviors.

The variation of False Acceptance and False Rejection Rates with α can be seen in figure 2. In table 2, some results obtained with this system are presented. These results are compared with the ones obtained with the Speaker Verification system. It can be observed that, in the best case (labeled Dual 3), the improvement in the FA and the FR rates are 7.1% and 23.8%, respectively.

Verification	α	FA Rate	FR Rate
Speaker	--	3.684%	8.077%
Dual 1	0.59	5.000% (+35.7%)	5.000% (-38.1%)
Dual 2	0.62	1.842% (-50.0%)	8.077% (±0.0%)
Dual 3	0.60	3.421% (-7.1%)	6.154% (-23.8%)

Table 2. Results obtained with the algebraic combination.

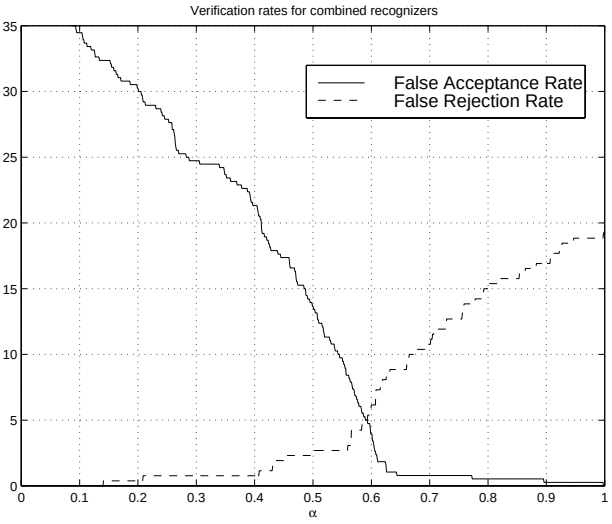

Figure 2. False Acceptance and False Rejection Rates obtained with the algebraic combination when α varies between 0 and 1.

4.2 Single Perceptron Combination

Taking a look to the O_{perm} definition, we see that its formulation is very similar to the output of a single-layer perceptron [6] in figure 3. Observe that at the input of the perceptron we have pairs $\boldsymbol{S}$ consisting of the scores S_u and S_s for each test.

$$y = Sig(\omega_s S_s + \omega_u S_u + \omega_0)$$

$$Sig(x) = \frac{1}{1+e^{-x}}$$

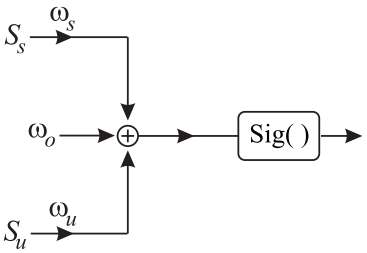

Figure 3. Single layer perceptron.

This is a classical two-class classification problem. For training the perceptron, we decided to use a cross-entropy error function. We have the data divided in two classes: class C_1 corresponds to the data in which the presumed and real identities are the same (customer tests) and C_2 is the opposite (impostor tests). The output y of the perceptron represents the posterior probability $P(C_1/\boldsymbol{S})$ for class C_1 while the posterior probability of class C_2 will be given by $P(C_2/\boldsymbol{S})=1-y$ where $\boldsymbol{S}=(S_u,S_s)$. This can be achieved if we consider a target coding scheme for which t=1 when the inputs S_u and S_s belong to class C_1 (a customer test) and t=0 when the inputs belong to C_2 (an impostor test). These two probabilities can be combined into a single expression so that the probability of observing either target value is

$$p(t/\boldsymbol{S}) = y^t (1-y)^{1-t}$$

With this interpretation, the likelihood of observing the training data set is then given by

$$\prod_{n=1}^{N} (y_n)^{t_n} (1-y_n)^{1-t_n}$$

where N is the total number of tokens. Due to numerical reasons, it is convenient to minimize the negative logarithm of this likelihood. This leads to the cross-entropy error function:

$$E = -\sum_{n=1}^{N} [t_n \ln(y_n) + (1-t_n)\ln(1-y_n)]$$

In this equation, the classification errors in customer (t_n=1) and impostor (t_n=0) tests are taken into account by left and right part of each term of the summation, respectively. Then, if we want to control the working point we must include a control parameter α:

$$E = -\sum_{n=1}^{N} [\alpha\ t_n \ln(y_n) + (1-\alpha)\ (1-t_n)\ln(1-y_n)]$$
$$0 \le \alpha \le 1$$

For small values of α, the system will tend to be restrictive and for values of α close to 1 the system will tend to be permissive. The final expression for the error function is:

$$E = -\frac{1}{N}\sum_{n=1}^{N}\left[\alpha\ P_{cu}\frac{N_{im}}{N} t_n \ln(y_n) + \mathrm{K}\right.$$
$$\left.\mathrm{K} + (1-\alpha)\ P_{im}\frac{N_{cu}}{N}(1-t_n)\ln(1-y_n)\right]$$
$$N = N_{im} + N_{cu}$$

The *1/N* factor is included because the training is performed in batch mode. N_{cu} and N_{im} are the number of customer and

impostor tests, respectively, in the training set. They are included in the error function to take into account the differences in the number of impostor and customer tests. Finally, there are two adjustable parameters, namely P_{cu} and P_{im}, in the error function. We experienced that if both parameters are set to 1, the resulting system becomes very restrictive. The reason for this behavior is that the perceptron is trained making use of scores obtained with the training material. Since these scores are quite high, particularly the true speaker scores S_s, the system is trained in such a way that it expects high values of them in case a customer test is being performed. But in the testing phase, these scores are not so high and many customers are rejected. According to a series of preliminary experiments we fixed P_{cu}=10 and P_{im}=0.1.

The set of parameters that define the perceptron are trained according to:

$$\left.\begin{aligned}(\omega_i)_{l+1} &= (\omega_i)_l + \Delta\omega_i \\ \Delta\omega_i &= -\lambda \frac{\partial E}{\partial \omega_i}\end{aligned}\right| \quad i = \{s,u,0\}$$
$$\text{until } (E_{l-1} - E_l) < \eta$$

with λ=0.25 and η=10^{-7}.

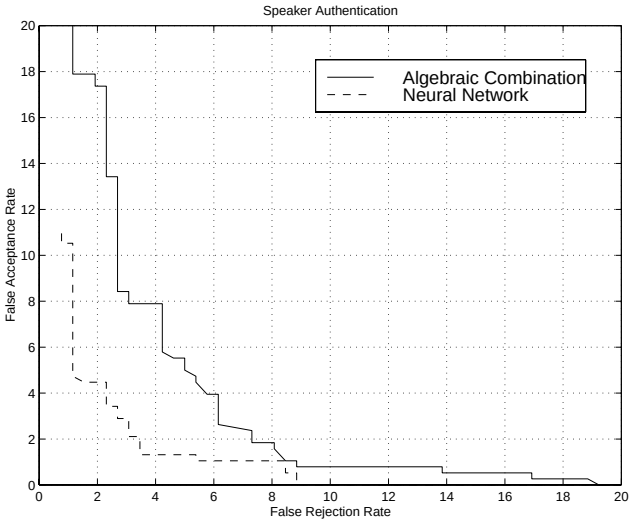

Figure 4. False Acceptance and False Rejection Rates obtained with the dual recognizers when α varies between 0 and 1.

Verification	α	FA Rate	FR Rate
Speaker	--	3.684%	8.077%
Dual NN	0.506	2.895% (-21.47%)	2.692% (-66.67%)

Table 3. Results obtained with the single perceptron.

The variation of False Acceptance and False Rejection Rates with α can be seen in figure 4 compared to the Speaker Verification using Algebraic Combination. In table 3, the best results obtained with this perceptron are presented compared to the Speaker Verification System. The FA and FR rates obtained with the system that makes use of a perceptron are both below 3%.

5. CONCLUSIONS AND FURTHER WORK

For comparison reasons, we include in table 4 the results presented all along this paper.

Verification	FA Rate	FR Rate
Speaker	3.684%	8.077%
Utterance	12.105%	21.923%
Dual 1	5.000%	5.000%
Dual 2	1.842%	8.077%
Dual 3	3.421%	6.154%
Dual NN	2.895%	2.692%

Table 4. Results obtained with the presented systems.

As it was expected, the performance of the classical GMM-based Speaker Verification systems can be improved with a Dual Verification system. We think that the use of verbal information is a practical and efficient way of overcoming the performance limits of the classical GMM architectures in Speaker Authentication tasks.

Besides, the performance of the Dual Verifier is dramatically improved by making use of a Neural Network structure due to its ability to learn from the data where to place the most suitable operating point. The results thus obtained encourage us to test more complex Neural Network architectures in the future.

One major drawback of the single perceptron combination consists in its dependence on the heuristic choice of some parameters like P_{cu} and P_{im}. We think that to divide the training material in two parts, one for training the models and the other for calculating thresholds and training the Neural Network, would be a good solution for this problem.

6. REFERENCES

[1] Rodríguez-Liñares L. and García-Mateo M., "On the Use of Acoustic Segmentation in Speaker Identification", *Proc. of EuroSpeech'97* 5: 2315, 1997.

[2] Reynolds, D. "Speaker identification and verification using Gaussian mixture speaker models", *Speech Communication* 17: 91-108, 1995.

[3] Rodríguez-Liñares L. and García-Mateo M, "A Novel Technique for the Combination of Utterance and Speaker Verification Systems in a Text-Dependent Speaker Verification Task", accepted for publication in the *Proc. of ICSLP'98* to be held in Sidney, Australia in Dec. 1998.

[4] Li Q., Juang B., Zhou Q. and Lee C., "Verbal Information Verification", *Proc. of EuroSpeech'97* 2: 839, 1997

[5] García-Mateo C. and Lee C., "A Study on Subword Modelling for Utterance Verification in Mexican Spanish", *Proc. of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding*, 614, 1997.

[6] Bishop, C., "Neural Networks for Pattern Recognition", Oxford University Press, 1995.