SPEAKER VERIFICATION ON THE WORLD WIDE WEB

Michael Sokolov

Digital Equipment Corporation
Cambridge Research Laboratory
1 Kendall Square, Cambridge, MA 02139 USA
Tel. (617) 692-7659, E-mail: sokolov@crl.dec.com
http://www.research.digital.com/CRL

ABSTRACT

This paper describes a system for controlling access to web resources built using well-known speaker verification techniques. We describe the implementation of a speech verification server and an associated authentication module for the Apache web server.

Speaker verification requires two inputs: a sample of the user's speech and an identity claim for the user; typically the user's name. However a more convenient system would not require a user name to be entered. We present the results of an attempt to implement speech-only authentication using open set speaker identification. We explore the effect of database size on performance.

INTRODUCTION

As the World Wide Web continues to expand, there is an increasing need for securing access to information published there. We believe that voice-based user authentication offers some unique benefits. Internet data publishers often complain that users exchange passwords freely, cicrumventing the publisher's attempts to charge for their services. Voice authentication cannot be given to another user. Furthermore, passwords may be forgotten; users cannot forget their own voices.

Using Digital's unique voice plug-in [1], we have implemented a speaker verification server for the World Wide Web, called *idserv*. This server, along with a special authentication module for the Apache web server (*q.v.* http://www.apache.org), forms the basis of a system for web-site user authentication based on voice. The enrollment portion of the system also functions as a more general speech data collection engine.

One earlier approach to web-based speech data collection used the telephone as a voice data transport [2]. Our system is unique in that it uses the "data" network for speech transmission. The idea of using a plug-in to mediate speech interfaces over the web was suggested in [3], although the authors opted for a different approach.

SYSTEM OVERVIEW

When users attempt to access one of the protected web pages at our site for the first time, they are presented with a page containing the voice plug-in. if they have not already done so, they are directed to download a copy of the voice plug-in. Then they enter their user name in a form on the web page and record a two-to-three second sample of their speech using the plug-in. The plug-in dispatches an HTTP POST request for verification. If verification is successful, the user is presented with the desired web page. The browser retains its authentication for several hours; thus protected pages may be visited thereafter without the need for any additional speech.

New users are also enrolled in our system using the same web interface. In order to maintain the integrity of the system, there must be some outside confirmation of a new user's identity when he or she is enrolled. One way to ensure this is to protect the enrollment pages with the access control system. Administrative users are granted access to these pages and can thus enable new users to enroll after verifying their identity in person.

The enrollment process begins with a page that prompts the user to enter biographical information, including a unique identifier, as well as information that characterizes the acoustic conditions (the microphone, the location). The user continues enrolling by recording approximately twenty to thirty seconds of speech. In order to ensure that we obtain a sufficient amount of speech, users are prompted with text to be read. We typically record ten short sentences, resulting in the desired twenty to thirty seconds of speech. Each utterance is checked to ensure that its dynamic range is within tolerance, and the duration of the utterance is also required to fall within a certain range. Users are prompted to re-record utterances that are either too soft or that are determined to have been clipped. The system records a word transcription file along with the speech data so that it may be used in a speech

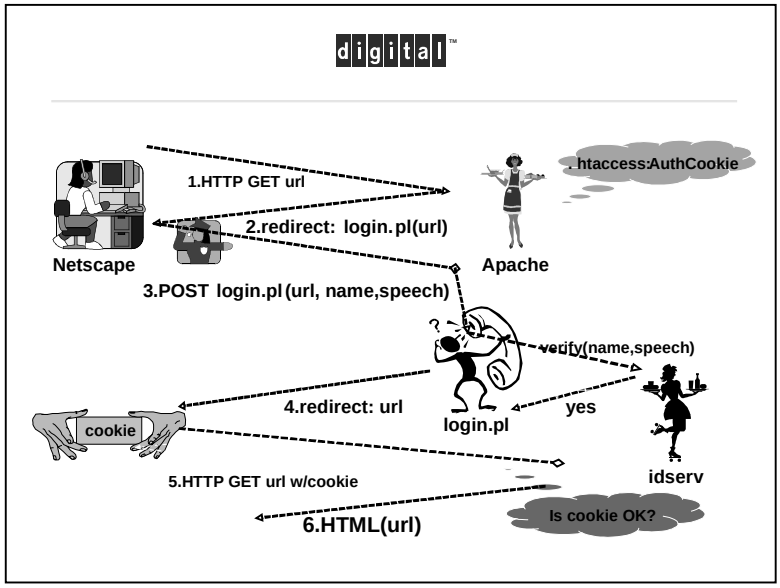

Figure 1. Web authentication schematic.

recognition database as well. Finally, when all the training sentences have been submitted, a speaker model is computed by *idserv*, and the user is registered in the speaker verification database.

ARCHITECTURE

Web servers can restrict access to their resources in a number of ways. Our system is built as a simple extension of the "basic" authentication style supported by all web servers and browsers. In this style, when a browser attempts to access a page that has been marked as accessible only to a particular user or group of users, the web server returns an error code that causes the browser to present the user with a name and password dialog. The browser then presents the name/password pair that the user entered whenever it accesses any resource on a branch off of the current page. Our system functions completely analogously to the "basic" system, but with a few important differences. Authentication is based on Netscape cookies, which are tokens stored in the client's browser. And the error code returned to the browser is a "redirect," rather than a request for user authentication. This requires only a small extension to the widely used and publicly available Apache server. In fact the cookie authentication mechanism simply provides a hook to an external program, whose job it is to verify the identity of the user and to provide the user with a cookie that will prove to the web server that they have been authenticated. In our case, this is a CGI script code (labeled *login.pl* in fig. 1) that invokes our speaker verification.

Netscape cookies are stored in a file, *cookies.txt*, on the client machine's hard drive. Because of this, we go to some trouble to ensure that a user seeking to break into the system cannot simply copy a cookie. Each cookie is encrypted using MD5 encryption, which is available on most platforms *via* the standard library function *crypt*. Each cookie contains a time stamp, the client's internet (IP) address, the user name used during authentication and a number that is unique to the web server granting authentication. Including all of these values protects against cookie copying as well as attempts to concoct cookies based on knowledge of the method used to generate them. Encrypting the speech data transmitted during authentication would provide additional security, as would the replacement of the MD5 encryption with a more secure variety.

ALGORITHMS

For each enrollment and each test utterance, we compute a set of Mel-frequency cepstral coefficients (MFCCs) similar to those used in many other systems. We have a high-pass pre-filter, followed by explicit DC bias removal and gain control (energy normalization), after which we compute 41 cepstral coefficients.

For efficiency, we may use fewer than 41 cepstra, but maximum accuracy is obtained with the full set. We then perform blind deconvolution by subtracting the mean from each of the cepstra in order to provide some degree of robustness to microphone and environmental variability. We have tried various techniques for modeling speakers, including AHS (for details see [4], [5]) and a GMM-based technique [6], and achieve performance comparable to that reported in the literature.

The core components of our system are not unique; the feature set and the scoring methods are well known. However, some considerations arise in the design of and use of a functioning system that suggest new avenues for exploration.

DESIGN CONSIDERATIONS

Automatic speaker recognition is extremely sensitive to noise and channel effects. Our own experiments, as well as others', (*e.g.* [7]) indicate that performance may be affected drastically by apparently minor changes in recording conditions. We believe that a major advance in basic performance will require a breakthrough in robustness to noise and other environmental effects. Without robustness processing, these effects can predominate in the statistical models used. Typically we encourage users to enroll in the system in an acoustic environment and with a microphone that corresponds to the way they will normally use the system. However, if a user attempts to gain access in a new environment or with some unfamiliar background noise present, they may well fail. In addition, it may be possible for impostors to gain entry to the system by simulating the conditions that existed when the true speaker enrolled. It is possible for a particular office environment to have such a unique spectral signature that false accesses will be allowed from that office simply on the basis of the environmental noise.

We see two main avenues of attack that seek to address this problem. The simpler, but perhaps more difficult approach is simply to collect a speech database with a very large number of speakers. Preliminary experiments indicate that overall system performance shows continuous improvement as the number of models available for impostor rejection is increased. A more complicated and less sure, but potentially much more rewarding approach is to develop a portable robustness processing method that would ameliorate the effects of background noises, reverberation, channel variations and other distortions.

We believe that the most valuable attribute of speech verification when compared to password verification is its convenience and naturalness. However, the current system requires users to enter a user name in order to be verified. We believe that eliminating this requirement will greatly enhance the users' perception of the simplicity and convenience of the system. To this end, we are pursuing methods of improving our open-set speaker identification performance.

EXPERIMENTS

We report results comparing open-set speaker verification, closed set verification, and open-set speaker identification. The easiest task is closed-set verification, in which all the speakers (including impostors) have models that are enrolled in the system's database, and in which a putative identity is provided with each test utterance. In the open set verification case, we assume that the impostors have never been seen by the system before; this is more representative of an actual security application. Finally, open set speaker identification asks the question "who is this," where the answer may be: "not anyone I know." This last case is the most interesting from the authentication point of view because if it worked, it would allow us build an access control system that would not require users to identify themselves *via* some alternative method such as entering a user name on a keyboard.

Setup

We performed experiments using the TIMIT database. In order to see the effect of changing database size, we used two sets: the full 630 speakers, and a 168-speaker subset. We trained our models with seven utterances and used the remaining three for testing. We verified test utterances by comparing scores obtained from the putative speaker's model with scores obtained from that speaker's cohort. In the open set verification, the true speaker's model was always excluded from the cohort. In closed set verification, it was not. To do open-set speaker identification, we just took the highest scoring model and compared its score against scores computed against that speaker's cohort. If the difference was greater than a chosen threshold, the speaker was accepted. This is equivalent to performing speaker verification against every model in the database at once, and or-ing the results together. The significant question here is not whether known speakers are identified correctly, but whether *unknown* speakers are confused with known ones.

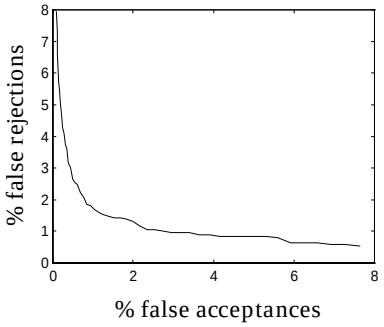

Figure 2 ROC for open-set verification on TIMIT's 630 speakers.

Results

Generally we think of identification results in terms of percent correct identification. Here we report equal error rates (which represent underlying ROCs, or *receiver operating curves*) for verification and for (open set) identification so that they can be compared directly. A typical ROC is shown in Figure 2. The curve shown there corresponds to the second row of Table 1. Each of the equal error rates reported in the tables below can be thought of as representing a similar ROC.

The first thing to note is that open set verification is harder than closed set verification, as we expected, and open set identification is the hardest, as we expected. What is more interesting though is how the relative difficulties change when we look at a larger data set. Open set performance improves (relative to closed set performance) when a larger data set is used. We hypothesized that this was the result of the increase in the number of models available for impostor rejection, or, put another way, better coverage of the feature space by the speaker cohorts, which are now drawn from a larger, more representative set of models.

Table 1 - equal error rates for TIMIT 168 speaker set

closed set verify	0.197
open set verify	1.23
open set identify	13.4

Table 2 - equal error rates for TIMIT 630 speaker set

closed set verify	0.877
open set verify	1.48
open set identify	16.4

To verify this hypothesis, we performed a second set of open set identification experiments that explored the idea of *garbage cohorts*. The idea is that if we maintain a large pool of speaker models from which to draw cohorts, we may be able to achieve better open set identification performance on smaller sets of actual candidate speakers. We did this by dividing our pool of models into two sets; one set was analogous to the set of speaker models used in the first experiments. The other set was the garbage speaker model set. Cohorts were drawn from either or both sets, but test utterances were

only identified as known if they could be identified as having come from one of the models in the non-garbage set.

Table 3 - garbage cohort experiment. Column headings indicate speaker pool size. Row headings indicate garbage cohort size.

	10	**20**	**40**	**100**
0	17.1	14.8	14.3	13.8
40	6.2	7.5	10.0	x
80	5.5	6.5	8.8	10.9*
200	3.7	5.3	6.7	9.9
400	3.6	4.7	5.7	8.3

*For this cell, garbage cohort size was 100.

Discussion

We found a marked and uniform improvement in system performance as we increased the size of the garbage cohort pool. For the smallest speaker pool size (ten), it looks as if the maximum benefit may have been achieved, as there was little improvement gained by doubling the garbage cohort size from 200 to 400. However, to be sure about these results, we believe that experiments with larger speaker pools need to be done.

A fundamental observation that remains to be explained is the dependence of open set identification performance on speaker pool size. In the absence of garbage cohorts (Row one of Table 3), performance improves as the speaker pool grows, up to a size of 168 (See Table 1). However, performance on the 630-speaker set has dropped down to the same level we would predict for set size of 15 based on the numbers in Table 3. We hypothesized that this might have to do with an increase in confusions of known speakers beginning to overwhelm the increase in rejection of impostors that derives from a more densely populated speaker space. If this is the case, this point might mark the point of diminishing returns for improved performance from increased database size.

In order to gain more insight into this question, we examined the number of false acceptances versus the number of false rejections.

It turns out that as we move from 168 to 630 speakers in the pool, the number of false rejections increases and the number of false acceptances decreases, for all values of the threshold parameter. It happens that the former is larger than the latter, and so overall performance decreases.

This observation does tend to indicate that the space is in some sense getting too crowded for our simple methods to work properly. We believe that the situation may be improved with alternative methods of cohort selection. However, we do not have adequate data to predict the performance of these systems on significantly larger databases, and it may be that simply throwing more speakers into a garbage population will ultimately be sufficient.

CONCLUSIONS

Speaker recognition is ideally suited for user authentication applications in which convenience is the most important consideration. Users no longer have to remember passwords. Widely held passwords do not have to be changed every time a potentially disgruntled employee departs. The hardware required is minimal and already widely available (in contrast with other biometrics, such as fingerprinting). Also, as with all biometric techniques, the token of identity is inseparable from the user; it cannot be lost or stolen. Text-dependent systems may guard against the use of recorded speech in situations where a higher level of security is required. There may be some deterrent effect if impostors know that their voice samples will be retained. However the level of accuracy achieved by current technologies makes this technique less suitable in situations in which a very high degree of security is required.

Furthermore, truly user-friendly and convenient speech-based authentication should not require any extra-modal identification, such as entering a user name. To this end, we investigated the open set identification task and found equal error rates about ten times greater than those for verification, when user names were provided. Results were improved using garbage cohorts, but it was not clear whether or not the maximal improvement had been achieved, due to a lack of available data.

REFERENCES

1. GOLDENTHAL AND WEIKART, Digital Web-based Speech Deployment and the Digital Voice Plug-in. Submitted *to Proc. EUROSPEECH 97*, Rhodes, 1997.
2. HURLEY, POLIFRONI, AND GLASS, Telephone Data Collection Using the World Wide Web, *Proc. ICSLP*, pp. 1898-1901, Philadelphia, 1996.
3. BAYER, S., Embedding Speech in Web Interfaces. *Proc. ICSLP*, Philadelphia, pp. 1684-1687, 1996.
4. GISH AND SCHMIDT, Text-Independent Speaker Identification. *IEEE Signal Processing*, pp.18-32, October 1994.
5. BIMBOT AND MATHAN, Text-Free Speaker Recognition using an Arithmetic-Harmonic Sphericity Measure, *Proc. Eurospeech 93*, pp. 169-172, Berlin, 1993.
6. REYNOLDS AND ROSE, Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models. *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 1, pp. 72-83, January 1995.
7. VAN VUUREN, S., Comparison of text-independent speaker recognition methods on telephone speech with acoustic mismatch. *Proc. ICSLP*, pp. 1788-1791, Philadelphia, 1996.