High Performance Connected Digit Recognition, Using Hidden Markov Models

S3.6

L. R. Rabiner, J. G. Wilpon, and F. K. Soong

AT&T Bell Laboratories
Murray Hill, NJ 07974

ABSTRACT — Algorithms for connected word recognition based on whole word reference patterns have become increasingly sophisticated and have been shown capable of achieving high recognition performance for small or syntax-constrained, moderate size vocabularies in a speaker trained mode. In this paper we use an enhanced analysis feature set consisting of both instantaneous and transitional spectral information and test the HMM-based connected digit recognizer in speaker trained, multi-speaker, and speaker independent modes. The performance that we achieved was 0.35, 1.65 and 1.75% string error rates for known length strings, for speaker trained, multi-speaker and speaker independent modes, respectively, and 0.78, 2.85 and 2.94% string error rate for unknown length strings for the 3 modes.

I. Introduction

The problem of recognizing strings of connected digits is crucial to a number of applications such as voice dialing of telephone numbers, and automatic credit card entry. In the last several years, several highly successful algorithms for recognizing spoken connected word strings from word prototypes have evolved [1-2]. These algorithms, all based on statistical pattern recognition methods, have achieved great success when applied to the problem of connected digit recognition. The reasons for this success are twofold; namely the fact that the recognition algorithms are optimal in the sense that they find the string of digit reference patterns that best (in some objective sense) matches the spoken digit string, and the development of highly successful training procedures which derive the digit reference patterns from a training set of fluent, connected, digit strings [3].

Earlier investigations showed that when a reasonable size training set was available for deriving the digit reference patterns, a fairly good recognizer could be implemented. The highest performance scores were achieved in a speaker trained mode; however performance was found to degrade seriously in either a multi-speaker or a speaker independent mode. Bush and Kopec found that by combining traditional pattern recognition techniques with acoustic-phonetic based rules, improved performance on speaker independent, connected digit recognition resulted [2].

In an effort to improve performance of the fully automatic connected digit recognition algorithms a major change was made in the front end spectral analysis. The analysis feature vector used for recognition, nominally an extended cepstral vector derived from LPC analysis, was augmented by the so-called delta cepstrum information [4].

The new analysis feature set was tested in the HMM-based connected digit recognizer in speaker trained, multi-speaker, and speaker independent modes, and was found to effectively reduce the string error rates by factors of 2 or more, often with considerably less computation than used previously. In particular, digit string error rates of 0.78%, 2.85%, and 2.94% were obtained for *unknown length* (UL) strings for speaker trained, multi-speaker and speaker independent tests, respectively. Comparable rates for *known length* (KL) strings were 0.35%, 1.65%, and 1.75%, respectively.

II. Review of HMM Connected Digit Recognizer

A block diagram of the overall level building, connected-digit recognizer is shown in Figure 1. There are essentially three steps in the recognition algorithm, namely:

(1) Spectral analysis — The speech signal, $s(n)$, is converted to a set of LPC derived cepstral (weighted) and delta-cepstral (weighted) vectors.

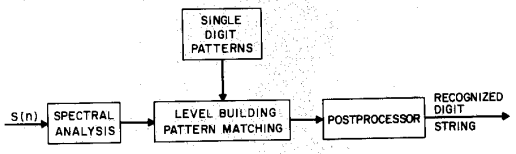

Figure 1 – Block diagram of connected digit recognizer.

(2) Level building pattern matching — The sequence of spectral vectors of the unknown speech signal is matched against a set of stored single-digit patterns (hidden Markov models) using the level building algorithm with Viterbi matching within levels. The output of this process is a set of candidate digit strings, generally of different lengths (i.e. different number of digits per string).

(3) Postprocessor — The output candidate strings from level building are subjected to further validity tests, e.g. state duration, to eliminate unreasonable candidates. The postprocessor chooses the most likely digit string from the remaining (valid) candidate strings.

In the remainder of this section we expand further on the LPC spectral analysis, and on the form of the HMM's. All other signal processing in the recognizer is essentially identical to that described in Reference 1.

II.1 LPC Spectral Analysis

The LPC front-end processing for recognition is shown in Figure 2. The overall system is a block processing model in which a frame of N samples is processed and a vector of features is computed. (Strictly speaking, as we will see below, this is not correct since the system uses a 5 frame window to compute the delta cepstrum vector.) The steps in the processing are as follows:

(1) Preemphasis — the digitized (at a 6.67 kHz rate) speech signal is processed by a first order digital network in order to spectrally flatten the signal.

(2) Blocking into frames — sections of N consecutive speech samples (we use $N = 300$ corresponding to 45 msec of signal) are used as a single frame. Consecutive frames are spaced M samples apart (we use $M = 100$ corresponding to 15 msec frame spacing, or 30 msec frame overlap).

(3) Frame windowing — each frame is multiplied by an N-sample window (we use a Hamming window) so as to minimize the adverse effects of chopping an N-sample section out of the speech signal.

(4) Autocorrelation analysis — each windowed set of speech samples is autocorrelated to give a set of $(p+1)$ coefficients, where p is the order of the desired LPC analysis (we use $p = 8$).

(5) LPC/cepstral analysis — for each frame, vectors of LPC coefficients are computed from the autocorrelation vector using a Levinson or a Durbin recursion method. The LPC derived cepstral vector is then computed up to the Q^{th} component, where $Q > p$, and $Q = 12$ in our implementation.

(6) Cepstral weighting — the Q-coefficient cepstral vector, $c_\ell(m)$,

CH2561-9/88/0000-0119 $1.00 © 1988 IEEE

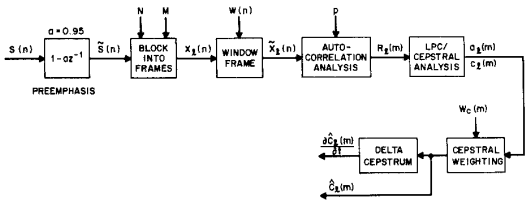

Figure 2 – Block diagram of improved front end LPC analysis incorporating instantaneous and transitional cepstral information.

at time frame ℓ is weighted by the window, $W_c(m)$, of the form [12,13]:

$$W_c(m) = \left[1 + \frac{Q}{2}\sin\left[\frac{\pi m}{Q}\right]\right], \quad 1 \leqslant m \leqslant Q \tag{1}$$

to give:

$$\hat{c}_\ell(m) = c_\ell(m) \cdot W_c(m) \tag{2}$$

(7) Delta cepstrum – the time derivative of the sequence of weighted cepstral vectors is approximated by a first order orthogonal polynomial over a finite length window of $(2K+1)$ frames, centered around the current vector. ($K = 2$ in our implementation; hence the derivative is computed from a 5 frame window.) The cepstral derivative (i.e. the delta cepstrum vector) is computed as

$$\Delta\hat{c}_\ell(m) = \left[\sum_{k=-K}^{K} k\,\hat{c}_{\ell-k}(m)\right] \cdot G, \quad 1 \leqslant m \leqslant Q \tag{3}$$

where G is a gain term so that the variances of $\hat{c}_\ell(m)$ and $\Delta\hat{c}_\ell(m)$ are about the same. (For our system the value of G was 0.375.)

The overall observation vector, $\mathbf{O}_\ell$, used for scoring the HMM's is the concatenation of the weighted cepstral vector, and the corresponding weighted delta cepstrum vector, i.e.

$$\mathbf{O}_\ell = \left\{\hat{c}_\ell(m),\ \Delta\hat{c}_\ell(m)\right\} \tag{4}$$

and consists of 24 coefficients per vector.

II.2 Hidden Markov Model Characterization of Words

Figure 3 shows the form of the HMM used to characterize individual digits. (Transitions between words are handled by a switch mode from the last state of one word model, to the first state of another word model, in the level building implementation.) The models are first order, left-to-right, Markov models with N states. (We have used values of N from 5 to 10.) Each state, j, is characterized by the following:

(1) A state transition vector, $\mathbf{a}_j$.

(2) A state observation density, $b_j(\mathbf{O})$, which is a continuous mixture density.

(3) Energy probability, $p_j(\epsilon)$, where ϵ is the dynamically normalized frame energy, and p_j is a non-parametric discrete density of energy values in state j obtained empirically from training data.

(4) State duration probability, $\hat{p}_j(\tau)$, where τ is the number of frames spent in state j, and $\hat{p}_j$ is an empirically measured, discrete density of duration values in state j.

In addition to the observation density, energy probability and state-duration probability, each HMM (for each word, v) is also characterized by an overall Gaussian word-duration density, $p_v(D)$.

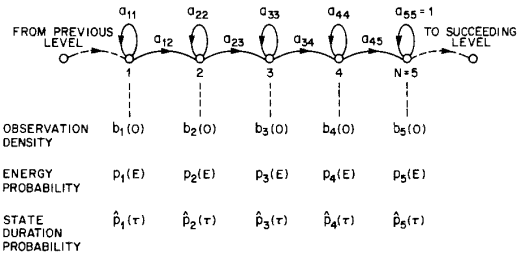

Figure 3 – Form of word HMM used to characterize individual digits.

III. Experimental Evaluation and Results

To evaluate the performance of the connected-digit recognizer, in speaker trained, multi-speaker, and speaker independent modes, two databases were used. The first database consisted of 50 talkers (25 male, 25 female) drawn from the local, non-technical, population (i.e. all talkers were native New Jersey residents). Each talker recorded 1200 connected-digit strings in about five sessions, during a 1-week period, over local dialed-up telephone lines. A new line was used for each recording session. The digits vocabulary consisted of the 10 digits (zero to nine); the word "oh" was excluded. Each talker recorded an equal number of strings with from 1 to 7 digits. Within each string the digits were selected at random; however during the test there was a constraint that there be an equal number of occurrences of each digit. All recordings were made in a reasonably quiet environment; however because of line variations and talker loudness variations, some recordings had very bad signal-to-noise ratios (i.e. on the order of 10-20 dB). A check was made on each recorded string to guarantee that the correct string was spoken. Because of the inexperience of the 50 talkers, a rather large number of the spoken strings were unusable (generally because of gross speaking errors in which only partial or incomplete strings were spoken), and about 21% of the 60,000 recorded strings (i.e. 12,600 strings) were eliminated. The talker with the most difficulty had about 50% of his strings (604 of 1200) eliminated; the talker with the least difficulty had only 47 of 1200 strings eliminated. Overall there remained 47,336 strings in the database. We denote the 50-talker database as DB50 in tables and in the text. This database was used in the speaker trained, and multi-speaker evaluations.

The second database, which was used to evaluate the connected digit recognizer in a speaker independent mode, was the TI connected digits database [5], as distributed by the National Bureau of Standards. This database contained connected digit strings from 225 adult talkers (equally distributed among male and female talkers), and was conveniently divided into training and testing sets, for consistency of comparison of results among the different researchers using this database. This database was dialectically balanced with an equal mix of talkers from 22 dialectical regions. At least 10 talkers (5 male, 5 female) from each dialectical region were included in the database. The vocabulary consisted of eleven words, namely the 10 digits and oh. Each talker spoke seventy-seven sequences of these digits, consisting of two tokens of each of the eleven digits in isolation, and 11 sequences of each of 2,3,4,5 and 7 digits (i.e. no 6-digit sequences were spoken). Digits were selected at random without replacement with one exception, namely the digits zero and oh never occurred in the same string. The digit strings were recorded in an acoustically treated sound room using a high quality microphone (Electro Voice RE-16 Dynamic Cardiod). All recorded strings were verified by a team of listeners at TI [5]. We refer to this database as DBTI in figures and in tables.

As provided by the National Bureau of Standards, the digitized strings were sampled at a 20 kHz rate. For consistency with the telephone bandwidth of the strings of DB50, all strings were digitally filtered to a 3.2 kHz bandwidth, and downsampled to a 6.67 kHz rate.

A total of 8568 training strings and 8578 testing strings were used (a small number of the strings on the digital tapes were unreadable). It should be noted that many of the strings had distinct silence gaps between groups of digits. Although it would have been possible to account for these silence gaps either by explicit methods (i.e. reendpoint the recorded strings) or by creating a silence model, neither of these procedures was actually used.

Database DB50 was split (at random) into a training set and a testing set, each consisting of roughly half the utterances for each talker and for each string length in the database. The training and testing sets for DBTI were specified by TI as an integral part of the database. The training sets were used to derive individual word HMM's; the independent test sets were used to measure system performance. The segmental k-means training procedure was always bootstrapped from word models derived from the isolated digits within the database [3].

III.1 Speaker Trained Mode Results

For the speaker trained case, an HMM with 8 states and 5 mixtures per state was used. The results of the recognition runs are given in Table 1. Table 1a gives string error rates (in %) for unknown length (UL) and known length (KL) strings, for both the training set and the independent testing set. Table 1b gives a breakdown of the string error rates for unknown length strings as a function of the number of digits in the string.

HMM	Training Set		Testing Set	
	UL	KL	UL	KL
8 states, 5 mixtures/state	0.39	0.16	0.78	0.35

HMM	Number of Digits in String						
	1	2	3	4	5	6	7
8 states, 5 mixtures/state	0.11	0.28	0.50	0.59	1.51	1.43	1.21

TABLE 1

The results show the following:

1. String error rates on the testing set are about twice as large as on the training set, although the absolute differences in error rates are still small.
2. String error rates for KL strings are about half those of UL strings for both the training and testing sets.
3. UL string error rates increase uniformly with the number of digits in the string, up to about 4 digits per string; for longer strings the error rates are much larger (around 1.4%), and are relatively insensitive to the number of digits in the string.

III.2 Multi-Speaker Mode Results

For the multi-speaker mode, using the training set of DB50, recognition systems were studied with from 1 to 6 models for each digit. The way in which multiple models were created was as follows. First, all the training strings were used to create a set of digit HMM's. (Two things should be noted here; first only one-fourth of the set of training strings were used, i.e. about 6000 strings, because of computational constraints in the clustering algorithms; second we only considered models with $N = 10$ states, $M = 9$ mixtures per state). Using a single model per digit set, (designed using standard methods), the 6000 training strings were optimally segmented into individual digits, and these digit tokens were clustered into from 1 to 6 clusters for each of the 10 digits. An individual HMM was designed for each of the clusterings, thereby leading to sets of HMM's with from 1 to 6 models per digit.

The results of the recognition tests in the multi-speaker mode are given in Figure 4 which shows string error rate for training and testing sets, as a function of the number of models per digit. The results show the following:

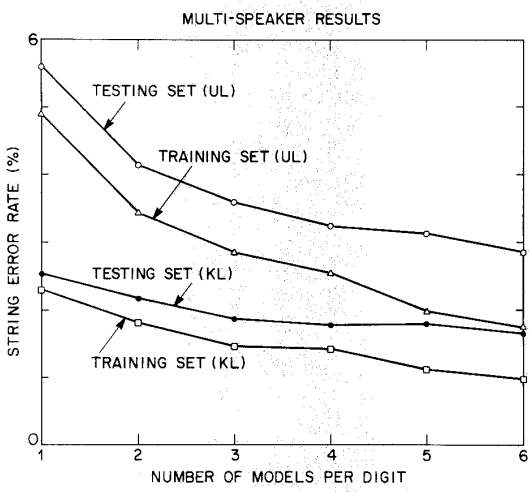

Figure 4 – String error rates as a function of the number of models per digit, multi-speaker case, for UL and KL strings, using both training and testing strings.

1. String error rates are significantly reduced by using more than 1 model per digit. For the training set, string error rates are reduced by a factor of about 2.5 as the number of models per digit goes from 1 to 6; for the testing set the reduction is about 1.7 to 1.
2. String error rates for training and testing sets are considerably closer than they were for the speaker trained case of Table 1.
3. For the case of 6 models per digit, the resulting string error rates on the independent test set were 2.85% for unknown length strings and 1.65% for known length strings.
4. The error rates for isolated digits are very low (0.22% for 6 models per digit); the string error rates rise uniformly for 2 to 5 digit strings, than even at a rate of about 4.5%.

III.3 Speaker Independent Mode Results

For the speaker independent tests of the recognizer, database DBTI was used. The specified training set was used to create from 1 to 6 models per digit, in a manner similar to the one used in the multi-speaker case. All 8565 training strings were used to create each set of models. The complete set of 8578 testing strings was used to evaluate the recognizer performance on the testing set.

The results of the speaker independent recognition tests are plotted in Figure 5 and show the following:

1. For the training set there is a reduction in string error rate of about 3 to 1 as the number of models per digit increases from 1 to 6; for the independent testing set the reduction in string error rate is only a factor of 1.5 for UL strings and 1.2 for KL strings.
2. A very large difference in performance exists between the training and testing sets, both for UL and KL strings. For example, for 6 models per digit, the string error rate for UL strings is a factor of 3 smaller; for KL strings the error rates differ by a factor of 5.5.
3. The string error rates on the testing set level off at about 3-4 models per digit; for 4 models per digit the UL string error rate is 2.94%, the KL string error rate is 1.75%.

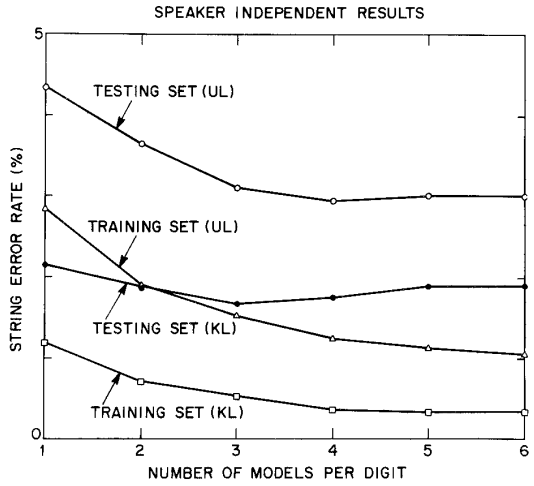

Figure 5 – String error rates as a function of the number of models per digit, speaker independent case, for UL and KL strings, using both training and testing strings.

4. The isolated digit error rate for 6 models per digit is 0.73%; string error rates for UL strings increase uniformly from 2 to 5 digits per string. For 7 digit strings, the string error rate is essentially equal to that of 5 digit strings since no possibility of digit insertions existed.

IV. Discussion

In this paper we have presented results that demonstrate major improvements in our ability to recognize unconstrained strings of connected digits. We have shown that by incorporating information about the time derivatives of the cepstral coefficients, along with instantaneous cepstral coefficients, we can significantly enhance recognizer performance. A summary of the recognizer performance, in each of the 3 modes in which it was tested, is given in Table 2. Overall string error rates of less than 3% for unknown length strings and less than 2% for known length strings were obtained on independent testing sets of data for both speaker independent and multi-speaker modes. String error rates of less than 1% for unknown length strings and less than 0.5% for known length strings were obtained in the speaker trained case.

Recognition Mode	Database	Training Set		Testing Set	
		UL	KL	UL	KL
Speaker Trained	DB50	0.39	0.16	0.78	0.35
Multi-speaker (6 Models Per Digit)	DB50	1.74	0.98	2.85	1.65
Speaker Independent (4 Models Per Digit)	DBTI	1.24	0.36	2.94	1.75

TABLE 2
Summary of String Error Rates for the Three Recognition Modes

These results show that the transitional cepstral information made the recognizer relatively robust to talkers. In independent work we have shown that the addition of the delta cepstrum analysis significantly improves performance with other vocabularies (e.g. the alphabet) in isolated word recognition tasks.

To see how much progress has been made, it is worthwhile contrasting the results presented here with those of earlier studies. In earlier work, using the same databases and recognizer, but with a standard instantaneous cepstral analysis (i.e. without the transitional cepstral information), Rabiner *et al.* reported testing set string error rates of 1.83% (UL), and 0.81% (KL) in the speaker trained mode, and 6.0% (UL) and 3.4% (KL) in the multi-speaker mode (using 10 models per digit as opposed to 6 models per digit here). The string error rates reported here are lower by a factor of 2 or more! Furthermore, in the speaker independent mode the results (reported at ICASSP 87 in Dallas) were testing string error rates of 7.9% (UL) and 5.2% (KL), again using 10 models per digit. Here the string error rates are lower by a factor of about 3 to 1, based on 4 models per digit. These comparisons strongly point out the advantages of the transitional cepstral information for recognition.

The only other comparison worth making is with the work of Bush and Kopec [2] who also used the TI database for their recognition tests. The best performance results on the testing set, obtained by Bush and Kopec, were 4% (UL) and 3% (KL) string error rates. The Bush and Kopec results were based on manually derived digit models (based on extensive manual analysis of the training set), using a wider bandwidth spectral analysis, with a network representation that handled difficult cases (e.g. prepausal oh or eight), and with an explicit background silence model. The results given here were obtained *fully automatically*, using telephone bandwidth data, with no explicit silence model, and with no rules or corrections for difficult digit sequences. All the techniques used here have been applied to several different recognition systems (different vocabularies, syntax etc.) with no modification whatsoever. This is almost as important and as impressive as the performance which has been demonstrated in this paper.

V. Summary

In this paper we have shown that a very high performance connected digit recognition system can be implemented automatically based on our current understanding. The key to the improvement in performance over earlier implementations was the use of an analysis that included both instantaneous and transitional (time derivative) spectral information. The system was tested in three modes, namely speaker trained, multispeaker, and speaker independent, and shown to be capable of recognizing digit strings with greater than 97% accuracy in all cases.

REFERENCES

[1] L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "A Model-Based Connected-Digit Recognition System Using Either Hidden Markov Models or Templates," *Computer Speech and Language*, Vol. 1, No. 2, pp. 167-197, Dec. 1986.

[2] M. A. Bush and G. E. Kopec, "Network-Based Connected Digit Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, Vol. ASSP-35, No. 10, pp. 1401-1413, Oct. 1987.

[3] L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "A Segmental *k*-Means Training Procedure for Connected Word Recognition Based on Whole Word Reference Patterns," *AT&T Technical Journal*, Vol. 65, No. 3, pp. 21-31, May/June 1986.

[4] F. K. Soong and A. E. Rosenberg, "On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition," *Proc. ICASSP 1986*, Paper 17.5.1, pp. 877-880, Tokyo, Japan, Apr. 1986.

[5] R. G. Leonard, "A Database for Speaker-Independent Digit Recognition," *Proc. 1984 ICASSP*, pp. 42.11.1-4, March 1984.