A Connected Digit Recognizer Based on Dynamic Time Warping and Isolated Digit Templates

L. R. Rabiner
C. E. Schmidt

Acoustics Research Department
Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

A connected digit recognizer is proposed in which a set of isolated word templates is used as reference patterns and an unconstrained dynamic time warping algorithm is used to literally "spot" the digits in the string. Segmentation boundaries between digits are obtained as the termination point of the dynamic path from the previous time warp. A region around the boundary is searched for the optimum starting point for the succeeding digit. At each stage the recognizer keeps track of a set of candidate digit strings for each test string. The string with the smallest accumulated distance is used as the preliminary string estimate. For variable length digit strings of from 2 to 5 digits (where the recognizer was not told the length of the string), word error rates of about 2-3% and string error rates on the order of 8% were obtained for both speaker dependent and speaker independent systems.

I. Introduction

Research in isolated word recognition has advanced to the stage where it is now possible to reliably recognize words from a vocabulary of up to several hundred words and phrases if the system has been trained to the talker [1-2]. For speaker independent recognizers, vocabularies of on the order of 50-100 words have been used and reliable recognition has been obtained by using multiple templates per word obtained from a clustering analysis of word tokens by a large set of talkers [3,4]. Although isolated word recognizers are suitable for a wide range of applications, for some important vocabularies the requirement for an isolated word format remains a major obstacle. An example of such a vocabulary is the set of digits (i.e. zero to nine).

In this paper we discuss a connected digit recognizer that has the following features:

1. It operates over dialed-up telephone lines.
2. It accepts variable length digit strings.
3. It can be used as either a speaker trained, or a speaker independent system.
4. It uses isolated word templates for the digits.
5. It uses an unconstrained dynamic time warping algorithm to segment and recognize the digits within the string.
6. It achieves high digit accuracy (98-99%) and high string accuracy (91-95%) for both male and female talkers.

II. The Connected Digit Recognizer

Figure 1 shows a block diagram of the connected digit recognition system. The analysis front end is similar to the one originally proposed by Itakura [2], and has been used in a wide variety of recognition experiments [3,4].

Following autocorrelation analysis, the endpoint detector finds the beginning and ending frames of the connected digit string, based on the log energy of the signal and the background silence statistics (as obtained during the recording interval) [3]. The pointer for the beginning of the string is set 5 frames before the indicated string beginning frame. A linear predictive coding (LPC) analysis is then performed (using a p=8 pole analysis) on each frame of the detected digit string. At this point we denote the test string as

$$T = T_1 T_2 \ldots T_L \tag{1}$$

where the frames T_j, $j = 1$ to L are p^{th} order, (normalized autocorrelation) feature vectors that describe the spectral properties of each frame of the digit string. If we denote the q^{th} reference pattern as

$$R^{(q)} = R_1^{(q)} R_2^{(q)} \cdots R_{M(q)}^{(q)} \tag{2}$$

where $R_i^{(q)}$ is the i^{th} feature vector (autocorrelated linear prediction coefficients) of the q^{th} reference pattern (of total length $M(q)$ frames), then the digit recognition problem is one of finding the optimum concatenation of reference strings to provide an optimum match between T and the concatenated string.

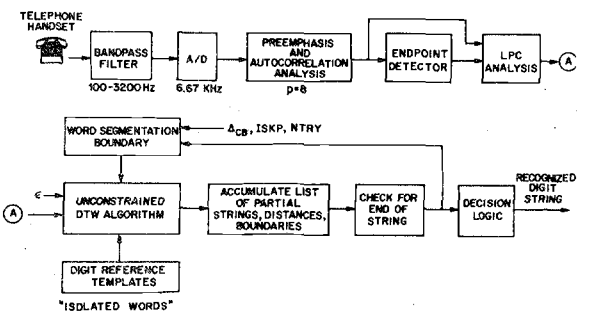

Fig. 1. Block diagram of connected digit recognizer.

We first define the frame distance between test frame j and reference frame i as

$$d(i,j) = \hat{d}(R_i, T_j) = \log(R_i \cdot T_j) \tag{3}$$

where T_j and R_i are $(p+1)^{st}$ order feature vectors, and $R_i \cdot T_j$ is a ratio of prediction residuals as defined by Itakura [2]. Following Sakoe [5], we now define the K-word concatenated reference string as

$$R^K = R^{(q(1))} \oplus R^{(q(2))} \oplus \cdots \oplus R^{(q(K))} \tag{4a}$$

$$= R_1^{q(1)} R_2^{q(1)} \cdots R_{M(q(1))}^{q(1)} R_1^{q(2)} \cdots \cdots R_{M(q(K))}^{q(K)} \tag{4b}$$

$$= \hat{R}_1 \hat{R}_2 \ldots \hat{R}_P \tag{4c}$$

where P is the total number of reference frames, i.e.

$$P = \sum_{k=1}^{K} M(q(k))$$

We can now state the optimum solution to the digit recognition problem as the sequence $q(k)$, $k = 1,2,\ldots,K$ that minimizes the quantity

$$D(R^K, T) = D(R^{(q(1))} \oplus R^{(q(2))} \oplus \cdots \oplus R^{(q(K))}, T) \tag{6}$$

over all possible K and $q(k)$. Figure 2a provides a pictorial representation of the digit matching process. If we segment T into K regions with beginning and ending points $b(q(k))$, $e(q(k))$, then the distance D of Eq. (6) can be decomposed into

$$D(R^K, T) = \sum_{k=1}^{K} \sum_{i=1}^{M(q(k))} \hat{d}(R_i^{q(k)}, T_{w(i)}) \tag{7}$$

where $w(i)$ is the optimum warping (to minimize the distance) between reference frame $R_l^{q(k)}$ and test frame $T_{w(i)}$ as obtained from a dynamic time warping algorithm. The beginning and ending functions b and e of the segmented test sequence T trivially obey the formulas

$$b(q(1)) = 1 \tag{8a}$$
$$b(q(k)) = e(q(k-1)) + 1 \tag{8b}$$
$$e(q(K)) = L \tag{8c}$$

To minimize the quantity of Eq. (6) over all k and $q(k)$ requires an inordinate amount of computation, even for modest values of K. For example, for $K = 3$ a total of $10^K = 1000$ digit strings are possible, and for each one a full dynamic time warping must be applied. (Recall that L for a K digit string is on the order of $40*K$ frames; hence a full dynamic time warping for each string requires about 1600 $K^2/3$ distances.) Furthermore, since K is apriori unknown, the computation must be carried out for each value of K that is anticipated (typically K=1 to 5). Clearly an exhaustive solution to the minimization problem is unfeasible.

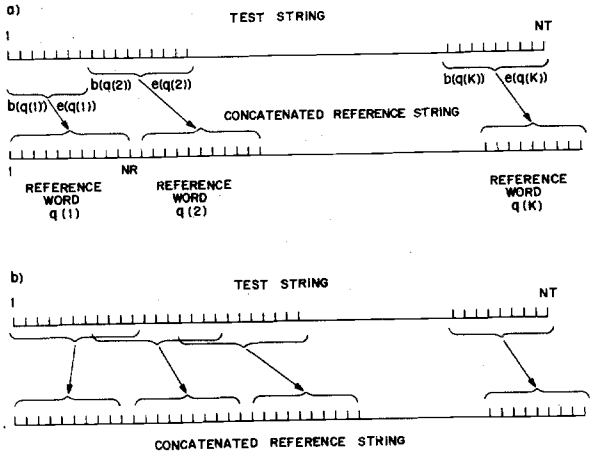

Fig. 2. Illustration of matching of concatenated reference string to test string. a) No overlap on test; b) with overlap on test.

The algorithm which was chosen to implement the string recognition is illustrated in Fig. 2b. The recognition of digits in the string is performed sequentially in time; however the endpoint constraints of Eq. (8) are not employed. Instead the k^{th} word is assumed to begin in a region around the end of the $(k-1)^{st}$ word, namely

$$e(q(k-1)) - \Delta_{CB} \leqslant b(q(k)) \leqslant e(q(k-1)) + 1 \tag{9}$$

where Δ_{CB} is the number of frames "cut back" from the end of the $(k-1)^{st}$ word. The overlap region between words accounts both for digit coarticulation, and for differences between isolated versions of a word and its properties when used in a connected format. The determination of K, the number of words in the string, is made from $e(q(k))$, the ending frame of the k^{th} word. K is chosen as the value of k such that

$$e(q(k)) \leqslant L - \Delta \tag{10}$$

where $\Delta = 10$ in our simulation, i.e. the last word must end within 10 frames of the end of the input string. Since a region is searched at the beginning of each word, the initial frame of the string was moved back by 5 frames to provide such a region for the first word in the string (and therefore keep the segmentation procedure algorithmic).

The operation of the recognizer is intimately tied to isolated word spotting for each word in the string. We are interested in investigating the region around $b(q(k))$ for the occurrence of word $q(k)$. Since word $q(k)$ can "begin" at any frame in the test frame, we need to evaluate the function

$$D(R^{q(k)}, T_l) = \frac{1}{M(q(k))} \sum_{i=1}^{M(q(k))} \hat{d}(R_i^{q(k)}, T_{w(i)}) \tag{11}$$

for $l = l_1,\ l_1 + IS,\ l_1 + 2IS, \cdots,\ l = l_1 + (NTRY-1)*IS$ where IS is the frame shift between estimated word beginnings, and $NTRY$ is the number of tries at a beginning point. A range of $NTRY*IS$ frames is used to find the best starting frame for the k^{th} digit in the string. However the dynamic time warping procedure itself provides a range of starting frames for the path. Hence a fairly wide range of test frames is searched to find the best estimate of the k^{th} digit. It should be noted that for "word-spotting" applications, values of $IS = 1$ are used - i.e. every single starting frame is searched to see if the word is present in the string. Hence here we are using a reduced sampling rate to try to spot digits in the string.

The recognition algorithm of Figure 1 (illustrated in Fig. 2b) proceeds from left to right. However, at each stage of the recognition, a list of partial strings, current boundaries, and distances is accumulated, as illustrated in Table I. For this example there were 2 candidates for the first digit (k=1), namely 5 and 9. Both candidates began at frame 6 of the test, but digit 5 matched until frame 35 whereas digit 9 matched until frame 24. The average distance (from Eq. (11)) for digit 5 was 0.44 whereas the average distance for digit 9 was 0.48, i.e. slightly larger. Since both digits were candidates for the first digit in the string, they were both retained as possible candidate strings. For the second digit (k=2), two sets of beginning frames were tried and, in both cases, the digit 0 best matched the input string. However a significantly better match was obtained in the vicinity of frame 24 than in the vicinity of frame 35; hence the most likely candidate string at the $k = 2$ position is the string 90 rather than 50. At this point the endpoint $e(2)$ for both strings is frame 72; hence there is no longer any possibility that partial string 50 will yield a smaller distance than partial string 90. At the third position (k=3) both partial strings find the digit 7 as the best match, with the digit 6 a somewhat poorer alternative. Hence an ordered list of 4 strings is obtained at this stage. Since all 4 partial strings end within a small number of frames of the digit string length, the recognition is terminated and the string 907 is chosen as the most likely candidate. However the ordered list of final candidates is retained for final testing using post correction techniques to be described later.

k	Partial String	Beginning Frame	Ending Frame	Accumulated Distance
1	5	6	35	0.44
	9	6	24	0.48
2	90	30	72	0.74
	50	20	72	0.80
3	907	70	107	1.02
	507	70	107	1.08
	906	68	109	1.25
	506	68	109	1.31

Table I

Typical Accumulation of Partial Digit Strings

The above example illustrates a number of points about the recognizer. These include:

1. Although a true 2 level dynamic warp is not used to make the final decision about the digits in the string, the algorithm used is reminiscent of the dynamic warping procedure and has the capability of choosing a partial string at stage k that had a higher distance at stage $(k-1)$ than other partial strings.

2. The overlap between the end of the $(k-1)^{st}$ digit and the beginning of the (k^{th}) digit is an important feature of the recognizer in that invariably a better digit match is obtained in the overlap region than would be obtained by tip-to-tip matching as required by other recognizers.
3. Although there is a tendency for the number of partial strings to increase with k (hence an increased computational load), it has been found that, in most cases, only a small number (often 1) of partial strings are retained.

In summary, the basic steps of the recognizer are:

1. Use an unconstrained dynamic time warping procedure to provide the best match to the first digit position. At the conclusion of the warping retain the partial string (or strings), beginning and ending frames in the digit string, and the accumulated distance scores.
2. Retain all partial strings (up to some maximum) that are either below a preset threshold, or are within a preset distance of each other.
3. For each partial k-digit string, obtain an updated $(k+1)$ digit string (or strings) whose accumulated distance is minimum by using an unconstrained dynamic time warping procedure to provide the best match in the vicinity of the ending frame of the k^{th} digit.
4. Check if the $(k+1)^{st}$ digit ends at or near the end of the test string. If not, repeat steps 2 and 3; if so the algorithm is finished.

III. Experimental Evaluation

To test the digit recognizer a series of recognition tests were run. First a preliminary evaluation was run on 2 talkers each speaking a randomized list of 10 strings of 3 digits each. For this preliminary run, parameters of the UELM algorithm were systematically varied (both for speaker trained and for speaker independent runs) and their effects on the recognition accuracy were investigated. Following this preliminary phase, a new series of recordings was made by 6 talkers (3 male, 3 female) who each spoke 80 strings of from 2 to 5 digits each.

3.1 Preliminary Investigation of UELM Parameters

A set of 10 strings of 3 digits each was used in the preliminary tests. The set of strings was recorded off a standard, dialed-up telephone line. For each of the 2 talkers, speaker dependent reference templates for the digits were obtained by having the talkers say each digit twice in a random, isolated sequence, and forming a reference template from each recording. Speaker independent templates were obtained from a clustering analysis of a 100 talker population [3].

The most significant parameters of the UELM dynamic time warping algorithm are the range parameters ϵ, and the cutback parameter Δ_{CB}. Figure 3 shows plots of contours of equal string error rate as a function of ϵ and Δ_{CB} for the recognition system of Section II for speaker dependent reference templates (Fig. 3a) and speaker independent reference templates (Fig. 3b). (Values of IS=3 and $NTRY$=4 were used in this and in many subsequent runs.) It can be seen from these figures that for a fairly large portion of the ϵ, Δ_{CB} plane, the string error rate remains 10% or less for both speaker trained and speaker independent templates. However it is seen that for this specific run, the regions of the (ϵ,Δ_{CB}) plane in which the minimum error occurs are fairly different. Thus the preliminary run only provided approximate regions of the (ϵ,Δ_{CB}) plane to test the recognizer for speaker trained and speaker independent digit templates. For the speaker trained case, it was anticipated that the optimum region in the plane could vary as more speakers were included in the test. However, for the speaker independent case, since the templates didn't vary, it was anticipated that the chosen region would be stable.

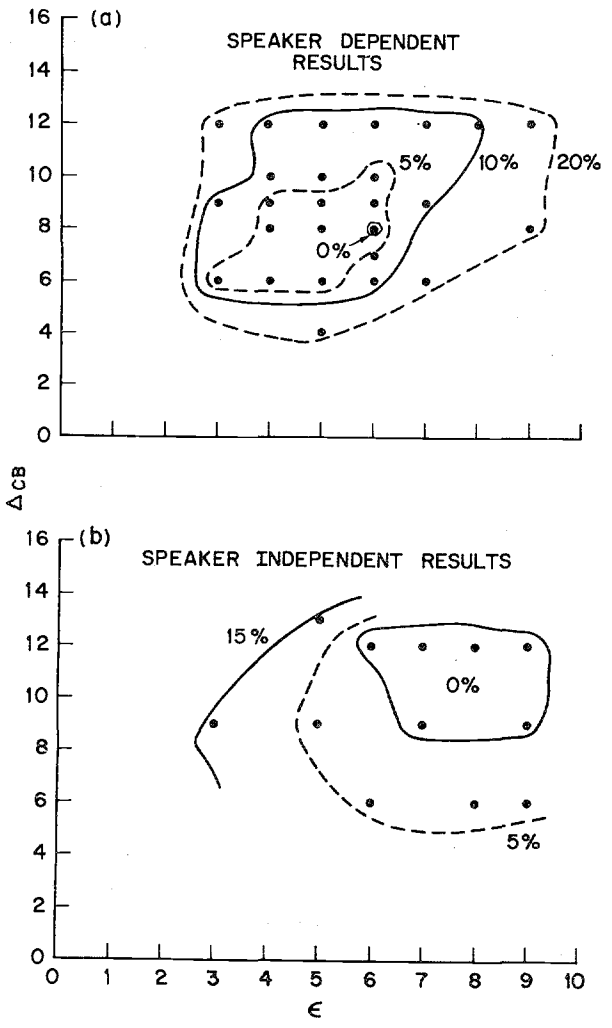

Fig. 3. Contour plots of recognition accuracy as a function of ϵ and Δ_{CB} for training sets of data for a) speaker dependent cases, and b) speaker independent cases.

3.2 Recognition Tests on Digit Strings

In order to test the recognition system, a randomized set of 80 strings of digits was used. The number of digits in the strings varied from 2 to 5 (20 strings each), and the number of times each digit appeared in each length string was uniform, but randomly generated. Each of six talkers (3 male, 3 female) spoke the strings over a dialed-up telephone line (a new line for each talker) and recorded the strings directly into the computer using a high speed array processor (the CSP MAP-200) for real time analysis and endpoint detection. As in the preliminary experiment, speaker dependent reference templates were obtained for each talker by having them recite each of the digits two times in an isolated word format. The speaker independent templates were again the clustered digits set as discussed previously.

The results of the recognition tests are given in Tables II and III, and in Figure 4. Table II gives results for the speaker dependent case for 2 sets of recognition parameters, and Table III gives results for the speaker independent case for 2 sets of recognition parameters. The data that is tabulated includes:

1. The number of string errors. A string error occurs when the best recognition candidate (with the lowest average distance) does not exactly match the input string. String errors can

occur because of digit errors, digit insertions, or digit deletions. For each talker a total of 80 strings were used; hence for each subject the number of string errors is relative to 80, and for the total count it is relative to 480.

2. The number of unfinished strings. A string was unfinished whenever one of two conditions was met. Either a digit insertion occurred in a 5 digit string causing the number of digits to exceed 5, or no reference digit was able to match the string features in the neighborhood of the ending point of the previous digit in the string. In either case the recognition was terminated with the partial match and no attempt was made to clear up the problem. (Unfinished strings were not counted as string errors; in fact in most cases the correct string was found.)
3. The number of digit errors. A digit error occurred whenever the wrong digit was recognized in place of the correct digit, at any place in the string where a digit truly occurred.
4. The number of insertions. A digit insertion was defined as the occurrence of an extraneous digit between two well defined digits in the string. This situation occurs in several well defined sequences - e.g. the introduction of a spurious 2 in the string 81 after the initial 8.
5. The number of deletions. A digit deletion was defined as the total absence of a digit which was actually spoken in the string with no subsequent replacement of that digit by another digit.

For the data of Table II (for the speaker trained system), it is seen that the results for the best set of recognition parameters from the preliminary run (Table IIa) were somewhat worse than the results for the recognition parameters set to values obtained for speaker independent recognition (Table IIb). A string error rate of 7.9% with a digit error rate of 1.3% was obtained for the speaker dependent system. A total of 8 digit insertions and 18 digit deletions occurred for these tests.

Talker	Number of String Errors	Number of Unfinished Strings	Number of Digit Errors	Number of Insertions	Number of Deletions
CS	2	1	2	2	0
LR	11	0	5	1	6
KS	9	0	7	1	3
SC	3	0	1	0	3
SL	9	0	4	0	7
JG	13	1	6	1	7
Totals	47	2	25	5	26
Percent Error	9.8	0.4	1.5		

(a) Speaker Dependent Recognition Results for $\epsilon = 6, \Delta_{CB} = 8, IS = 3, NTRY = 4$

Talker	Number of String Errors	Number of Unfinished Strings	Number of Digit Errors	Number of Insertions	Number of Deletions
CS	4	2	3	2	0
LR	5	0	3	1	2
KS	5	0	5	0	1
SC	3	0	2	0	2
SL	10	1	3	3	8
JG	11	0	6	2	5
Totals	38	3	22	8	18
Percent Error	7.9	0.6	1.3		

(b) Speaker Dependent Recognition Results for $\epsilon = 8, \Delta_{CB} = 12, IS = 3, NTRY = 4$

Table II

Talker	Number of String Errors	Number of Unfinished Strings	Number of Digit Errors	Number of Insertions	Number of Deletions
CS	8	2	4	8	1
LR	12	0	9	1	3
KS	2	0	2	0	1
SC	11	0	13	0	0
SL	12	0	7	0	7
JG	9	0	10	2	0
Totals	54	2	45	11	12
Percent Error	11.3	0.4	2.7		

(a) Speaker Independent Recognition Results for $\epsilon = 8, \Delta_{CB} = 12, IS = 3, NTRY = 4, KNN = 2$

Talker	Number of String Errors	Number of Unfinished Strings	Number of Digit Errors	Number of Insertions	Number of Deletions
CS	10	2	6	9	0
LR	10	1	12	2	0
KS	8	2	3	4	4
SC	12	1	9	5	1
SL	17	0	10	0	8
JG	6	0	6	1	2
Totals	63	6	46	21	15
Percent Error	13.1	1.3	2.7		

(b) Speaker Independent Recognition Results for $\epsilon = 8, \Delta_{CB} = 12, IS = 3, NTRY = 4, KNN = 1$

Table III

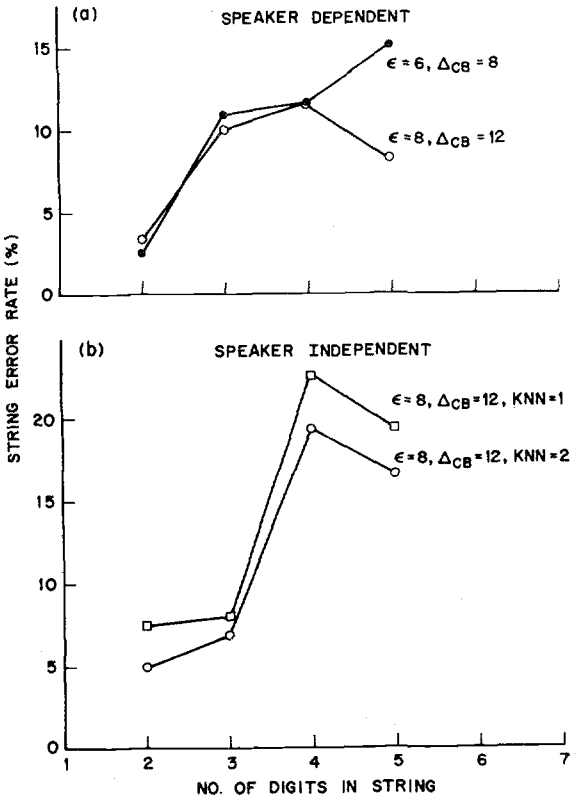

Fig. 4. Plots of average string error rate for a) speaker dependent results, and b) speaker independent results.

For the speaker independent recognizer the error rates were somewhat higher than for the speaker trained recognizer, as seen in Table III. For these cases the K-nearest neighbor recognition rule [3] was exploited to improve recognition accuracy by using $KNN = 2$ (average of 2 best templates) instead of $KNN = 1$ (minimum distance rule). For this case the average string error rate was 11.3%, with a digit error rate of 2.7%, and with 11 insertions and 12 deletions occurring in the tests.

Figure 4 shows plots of the string error rate as a function of the number of digits in the input string for the test cases of Tables II and III. Figure 4a is for the speaker dependent recognizer; whereas Fig. 4b is for the speaker independent case. It can be seen that for the 2 digit sequences, the string error rate is 2.5% for the speaker trained system and 5% for the speaker independent system. For 3 digit sequences there is a sharp increase in error rate to 10% for the speaker trained system; however for the speaker independent recognizer the error rate rises only to about 6.5%. For 4 and 5 digit strings, there are only small changes in error rate for the speaker trained system; however a very sharp increase in error rate occurs for the speaker independent system. As the number of digits in the string increases, the average time per digit tends to decrease - i.e. the talkers speak more rapidly. The results of Figure 4 indicate that the speaker independent recognizer can handle speaking rates corresponding to 3 digit strings or less with reasonably low error rates; beyond this point a sharp breakdown in accuracy occurs. On the other hand the speaker dependent recognizer tends to degrade more gracefully as the number of digits in the string increases. In fact for 5 digit strings the average error rate was smaller than for 3 digit strings.

One final point worth noting about the errors in recognition concerns the digit errors, the digit insertions and digit deletions that occur. For the speaker dependent system, the digit errors occurred uniformly across all digits; however the digit insertions occurred primarily for the digit 8, and the digit deletions occurred primarily for the digits 2 and 8. For the speaker independent case about half the digit errors occurred for the digit 2. The vast majority of the digit insertions and deletions were for the digits 2 and 8. It was anticipated that the digits 2 and 8 would experience the worst recognition problems because they are short digits which are heavily coarticulated and which can readily be deleted or inserted in connected strings. It was also expected that the problems with the digits 2 and 8 would be more severe for the speaker independent system, since the variability across the 12 templates was much larger than the variability across 2 speaker specific templates.

IV. Summary

In this paper we have discussed a system for the recognition of strings of connected digits over dialed-up telephone lines. The string length is unspecified and is determined by the recognition algorithm. The system uses isolated word templates as the basis of a pattern matching algorithm and can be used as either a speaker trained, or a speaker independent recognizer, depending on the set of templates.

The recognition process can be viewed as a modified form of the two-stage dynamic programming procedure proposed by Sakoe and used in the NEC hardware recognizer. The digits in the string are recognized sequentially using an unconstrained dynamic time warping algorithm and then a region around the endpoint of the k^{th} digit is used as the beginning region of the $(k+1)^{st}$ digit. The output of the recognizer is an ordered set of candidate strings of digits, ordered by average distance.

Although not described here, post correction process was then used to provide a small improvement in the recognition accuracy based on whole string comparisons using a constrained dynamic time warping algorithm. The recognition system obtained digit accuracies of about 97-99%, and overall string accuracies (after post correction) of from 91 to 93%.

References

[1] T. B. Martin, "Practical Applications of Voice Input to Machines," *Proc. IEEE*, Vol. 64, pp. 487-501, Apr. 1976.

[2] F. Itakura, "Minimum Prediction Residual Applied to Speech Recognition," *IEEE Trans. Acoustics, Speech, and Signal Proc.*, Vol. ASSP-23, pp. 67-72, Feb. 1975.

[3] L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, Vol. ASSP-27, No. 4, pp. 236-349, Aug. 1979.

[4] L. R. Rabiner, and J. G. Wilpon, "Speaker Independent, Isolated Word Recognition for a Moderate Size (54 word) Vocabulary," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, Vol. ASSP-27, No. 6, Dec. 1979.

[5] H. Sakoe, "Two Level DP-Matching-A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, Vol. ASSP-27, No. 6, Dec. 1979.