A Performance Evaluation of a Connected Digit Recognizer

L. R. Rabiner
J. G. Wilpon
B. H. Juang

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT. In this paper we discuss a system for automatically recognizing fluently spoken digit strings based on whole word reference units. The system that we will describe can use either hidden Markov model (HMM) technology or template-based technology. The training procedure derives the digit reference patterns (either templates or statistical models) from connected digit strings. To evaluate the performance of the overall connected digit recognizer, a set of 50 people (25 men, 25 women), from the non-technical local population, was each asked to record 1200 random connected digit strings over local dialed-up telephone lines. Both a speaker trained and a multi-speaker training set was created, and a full performance evaluation was made. Results show that the average string accuracy for unknown and known length strings, in the speaker trained mode, was 98% and 99% respectively; in the multi-speaker mode the average string accuracies were 94% and 96.6% respectively.

I. Introduction

One of the most important problems in speech recognition is connected digit recognition. Connected digit recognizers have significant applications in the area of telecommunications, as well as for recognizing spoken credit card numbers, stock codes, etc. For the applications above, a speaker independent system would generally be required. However there are a wide range of applications for speaker trained connected digit recognizers, including specialized operator services, insurance claims entry, quality control, package handling and sorting, etc.

Because of its vast potential applications, a wide variety of approaches to connected digit recognition have been proposed and evaluated [1-5]. One of the most interesting aspects of the connected digit recognition problem is that *whole word* training patterns can be used as the basic speech recognition unit to find the best matching string. Hence all the technology and research associated with whole word speech recognition can be brought to bear on this problem.

As with any pattern recognition algorithm, a major factor in determining the performance of the algorithm is the manner in which the reference patterns for the system are derived. For connected digit recognition, a training algorithm was developed in which the digit reference patterns were derived from fluent connected digit strings using a segmental k-means algorithm to split the connected strings into individual digits [4]. This training procedure was integrated into a level building, connected digit recognizer and tested on 50 naive talkers (25 male, 25 female), who were recruited from the local non-technical population. Both speaker trained and multi-speaker recognition tests were performed.

II. The Overall Level Building, 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 either a set of LPC vectors, or a set of cepstral vectors.
2. Level Building Pattern Matching - The sequence of spectral vectors of the unknown speech signal is matched against stored single digit patterns (either templates or statistical models) using the level building algorithm. The output of this process is a set of candidate digit strings, generally of different lengths (i.e., different number of digits per string).

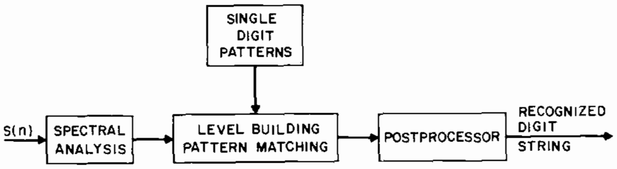

Fig. 1 Block diagram of overall connected digit recognizer.

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

In this work we have considered both template-based and statistical model-based (hidden Markov model, HMM) systems. The details of the LPC front end, the level building pattern matching, and the postprocessor are given in Ref. [4] and will not be repeated here.

III. Experimental Evaluation of Connected Digit Recognizer

The speech database consisted of 50 talkers (25 male, 25 female) drawn from the local, non-technical, population (i.e., all talkers were local New Jersey residents). Each talker recorded 1200 connected digit strings in about 5 sessions, during a 1 week period, over local dialed-up telephone lines. A new line was used for each recording session. 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, and that no gross endpoint errors were made. Because of the inexperience of the 50 talkers, a rather large number of the spoken strings were unusable (generally because of gross speaking errors), 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.

To get an idea of the average rate at which the digit strings were spoken, Figure 2 shows a plot of the average speaking rate (words per minute, wpm) versus the number of digits per string, for DB50 with data from 7 other talkers included. The average rate for isolated digits was about 137 wpm; the rate gradually rises to about 170 wpm and remains there for strings of length 4-7 digits. By contrast we have also plotted the wpm curve for the TI database of connected digit strings [6] (only adult male and female talkers were used for the TI curve). It can be seen that the talking rates of the TI talkers were somewhat slower (about 10 wpm) than those of the 57 talkers used here). This difference in rate could be accounted for by the fact that sometimes there were internal pauses (silence regions) in the TI database and these were not compensated in the rate calculations.

The database, DB50, was split (at random) into a training set and a testing set, each consisting of roughly half the utterances for each talker in the database. The training set was used to derive reference

CH2396-0/87/0000-0101 $1.00 © 1987 IEEE

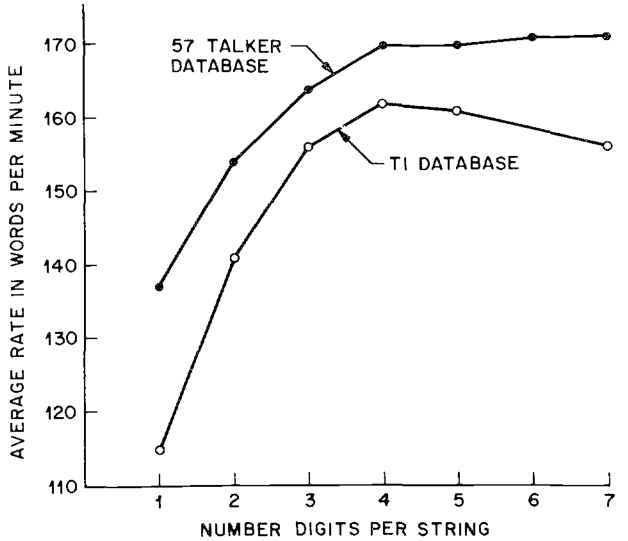

Fig. 2 Curves of the average speaking rates (in words per minute) as a function of the number of digits in the string.

patterns (either an HMM or a set of reference templates); the independent test set was used to evaluate the system performance. For the speaker trained system, the segmental k-means algorithm was always bootstrapped from a speaker independent set of templates or models. For the multispeaker system, the segmental k-means algorithm was bootstrapped either from speaker trained HMM's, or from a uniform state segmentation on the isolated digits within the database. Both HMM"s and templates were created in all cases and evaluated on the independent test sets.

3.1 Speaker Dependent Results - DB50

In the case of the speaker trained HMM recognizer, an $N = 5$ state, $M = 3$ mixture, $D = 8$ dimensional (cepstral vector without the zeroth order term) model was used for DB50 evaluations. None of the 3 model parameters (N, M, D) was varied since, as will be shown, the recognition performance was exceptionally good and it would have been difficult to assess whether changing a particular model parameter actually improved performance.

For each talker in DB50 a segmental k-means loop was run 10 times on each training set. At the end of each iteration an HMM was obtained and used to evaluate overall recognition performance. Similarly at the end of each training loop, a clustering procedure was used to cluster all the training tokens for each digit into a 3-cluster solution (i.e., 3 templates), for the template-based recognizer. The resulting HMM, at each iteration, for each talker, was run on both the training and testing strings, and the results of these evaluations are given in Table I. This table has 3 parts. In part a, the string error rates (%) for Unknown Length (UL), Known Length (KL), and for strings in which No Match (NM) among *any* of the candidates was found, for both training and testing strings are given. Part b of the table shows string error rates for UL strings as a function of the best candidate position in the ordered list of strings at the output of the recognizer. Thus the string error rate at best candidate position 3 is the percentage of strings which were not in the top 3 candidates. Finally, part c of the table shows string error rate, for UL and KL strings, as a function of the k-means iteration. This part of the table is somewhat deceptive since some talkers converged very rapidly (i.e., 1-3 iterations), while others converged more slowly. After convergence, the performance scores tended to vary by about ±1/2%, due to the tendency to capture details in the training set which did not occur in the testing set.

The overall results on DB50 show the following:

String Length	Training			Testing		
	UL	KL	NM	UL	KL	NM
1	0.14	0.03	0.03	0.95	0.19	0.11
2	0.43	0.09	0.03	0.85	0.28	0.09
3	0.83	0.42	0.12	1.65	0.68	0.15
4	0.77	0.54	0.24	2.07	0.89	0.35
5	1.37	0.70	0.21	2.38	1.03	0.45
6	1.41	0.41	0.19	2.79	1.05	0.34
7	0.84	0.44	0.22	2.30	1.52	0.71
Average	0.81	0.36	0.14	1.83	0.79	0.31

(a) String Error Rates (%) for UL, KL, NM Strings for DB50 (Best iteration for each talker)

Evaluation	Best Candidate Position				
Set	1	2	3	4	5
Training	0.81	0.23	0.15	0.14	0.14
Testing	1.83	0.66	0.39	0.34	0.31

(b) String Error Rates (%) for UL Strings as a Function of Best Candidate Position for DB50

Evaluation	Iteration									
Set	1	2	3	4	5	6	7	8	9	10
Training-UL	2.04	1.34	1.20	1.14	1.13	1.06	1.08	1.03	0.99	1.00
Training-KL	0.90	0.62	0.56	0.55	0.53	0.49	0.50	0.48	0.45	0.45
Testing-UL	3.09	2.52	2.24	2.18	2.21	2.23	2.24	2.21	2.19	2.10
Testing-KL	1.58	1.18	1.11	1.04	1.06	1.01	1.01	1.01	1.01	0.98

(c) String Error Rate (%) for UL and KL Strings as a Function of Iteration for DB50

TABLE I

1. The string error rates for both UL and KL strings, for both training and testing, are very low, and represent the best performance obtained to date on any connected digit recognizer.
2. There is a small, but consistent, difference in performance between training and testing sets.
3. The error rate for best candidate position 2, for UL strings, is essentially negligible, indicating that in cases when the recognizer made an error, the correct string was almost always the second choice.
4. A sharp improvement in performance is obtained from iteration 1 to iteration 2 in the segmental k-means training loop; a somewhat smaller improvement is obtained on the third iteration. Beyond this point, on average, the algorithm converged and the differences in performance were essentially statistical in nature.

The performance of the (3 template per digit) template-based recognizer was slightly worse than that of the HMM-based system (3.0% for UL strings, 1.6% for KL strings) but still was very good.

3.2 Multi-Speaker Results - DB50

The results of the performance evaluation on DB50, in the multi-speaker mode, using the HMM-based recognizer are given in Table II. The way in which the multi-speaker HMM's were derived is as follows. Some preliminary experimentation was performed to determine good choices for the number of states in the model, N, the number of mixtures per state, M, and the parameter set for the mixtures. It was found that models with $N = 8$ states performed somewhat better than models with $N = 5$ states. To demonstrate the effects of mixture parameters on performance, Figure 3 shows a set of curves of the average string error rate as a function of the number of mixtures per state for UL strings (part a) and KL strings (part b), for three parameter representations, namely:

1. 8 unweighted cepstral coefficients with diagonal covariance matrices (UCC/DC)
2. 8 unweighted cepstral coefficients with full covariance matrixes (UCC/FC)
3. 12 weighted cepstral coefficients with diagonal covariance matrices (WCC/DC)

(The case of 12 weighted cepstral coefficients with full covariance matrices was also tried but the performance scores were significantly worse than those shown in Figure 3. The poor performance scores were due to two factors. First, the weighting tends to decorrelate the cepstral coefficients; hence the off diagonal terms are very noisy. Second, the number of coefficients per mixture that needed to be estimated was large (144) and there was not sufficient data for good estimates of all these coefficients).

The curves in Figure 3 show the following:

1. For a large number of mixtures per state (6-9) the best performance came when using the set of 12 weighted cepstral coefficients with a diagonal covariance representation. The set of 8 unweighted cepstral coefficients with a diagonal covariance was the next best parameter set, with the full covariance representation of the 8 unweighted cepstral coefficients giving the worst performance. This result is undoubltedly due to the difficulty of reliably estimating the off diagonal terms of the covariance matrices, when the number of mixtures is large (i.e. with a small amount of data per mixture).
2. For a small number of mixtures per state (1-2) the best performance came from using the set of 8 unweighted cepstral coefficients with a full covariance representation. Here the off diagonal terms have enough training data to give fairly reliable estimates of their correct values.

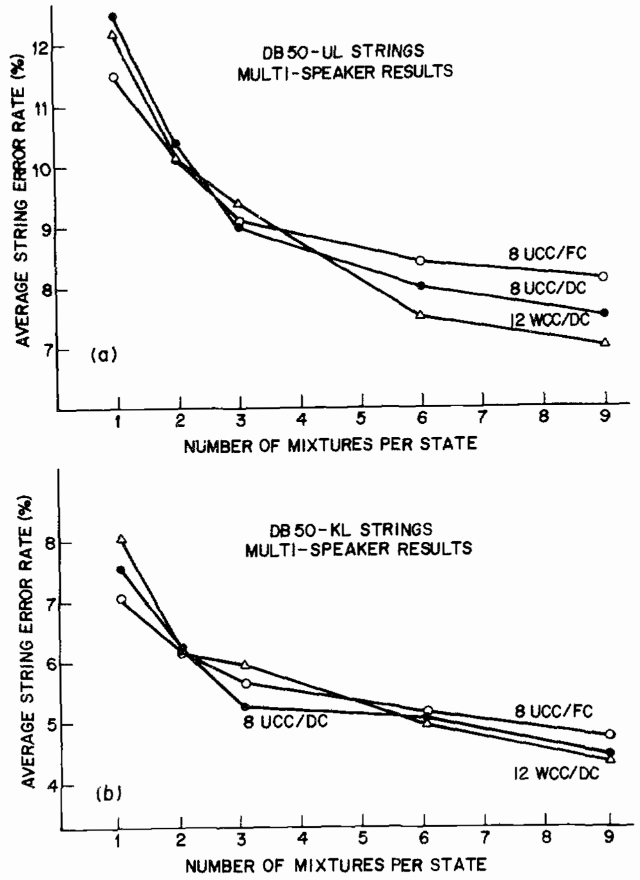

Fig. 3 Plots of average string error rate as a function of the number of mixtures per state.

Based on the results shown in Figure 3, we used a value of $M = 9$ mixtures per state with the 12 weighted cepstral coefficients as the best representation for connected digit recognition.

To bootstrap the models, we initially used the isolated digit sequences from all 50 talkers in the training set. Each digit was initially linearly segmented into states, and the segmental k-means training loop was run to convergence. A performance evaluation was run on an independent testing set consisting of 3500 random strings (10 strings of each of 1-7 digits for each of the 50 talkers, derived from the independent testing sets), and this result is shown as the "Isolated Digits" training condition in Table II. It can be seen that for this condition the string error rates are quite high (26.1% for UL strings, 18.5% for KL strings). This result is not unexpected since we have shown the inadequacy of isolated digit training previously.

The second step in building digit HMM's is to run the segmental k-means training loop using the isolated digit model as the initial model estimate. Because of the large size of the training set, only 1 of each 4 strings, for each talker, was actually used. This gave about 6000 training strings with about 24000 digits. The results for the full training set, using a single HMM per digit, are shown as the second line in Table II. String error rates improve by about 10% for both UL and KL strings.

The third and last step in building digit HMM's is to cluster the segmented digit tokens and to build a separate HMM for each cluster and for each digit. In this manner it is possible to build any number of HMM's for each digit. Figure 4 shows the behavior of the average string error rate as a function of the number of models per digit (as obtained from the clustering analysis) for both UL and KL strings. (These results were obtained using the 8 UCC/DC representation). It can be seen that substantial performance improvements are obtained as the number of models per digit increases from 1 to 5; an increase to 10 models per digit leads to a smaller performance improvement. Further

	String Error Rate (%)		
Training Condition	UL	KL	NM
Isolated Digits	26.1	18.5	8.7
Full Training - No Clustering - 1 Model/Digit	16.1	9.0	2.1
Full Training - Clustering - 10 Models/Digit	6.0	3.4	0.5

TABLE II

String Error Rates for DB50 for Multi-Speaker Training Using HMM Models

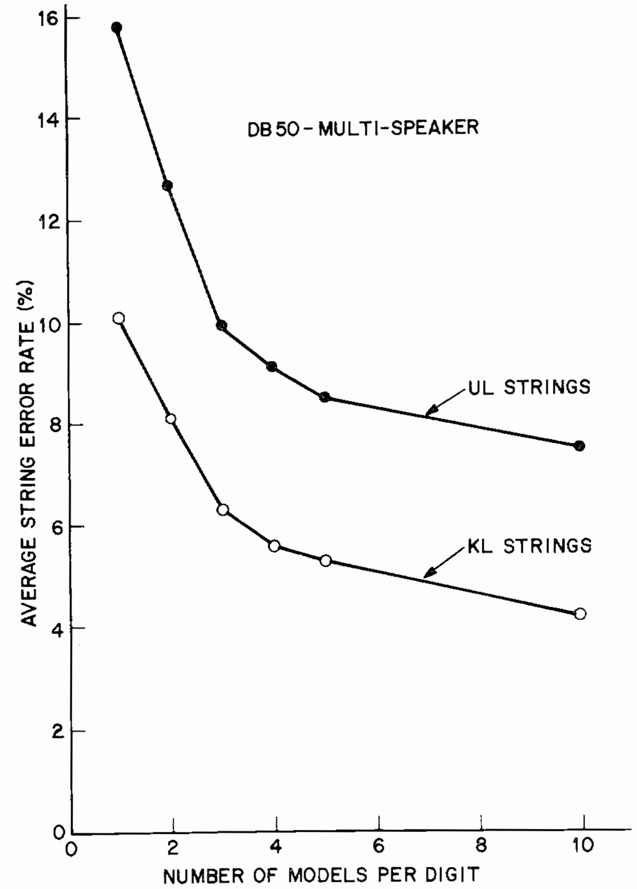

Fig. 4 Plots of average string error rate versus the number of models per digit.

increases in the number of models per digit (to 15 and 20) led to worse performance as the amount of training data per model is substantially reduced, leading to pooer estimation of model coefficients. The results in Table II are given for the case of using 10 HMM's per digit when a final stage of Baum-Welch reestimation of all model parameters is used. The average string error rate falls to 6.0% for UL strings, and to 3.4% for KL strings. Furthermore the percentage of strings for each no match is found falls to 0.5%, an acceptably small number.

Although the error rates for the multispeaker case are considerably larger than for the speaker trained case, they are substantially better than those obtained from earlier training methods. For example we tried a true speaker independent set of digit HMM's derived from embedded training methods, and measured UL string error rates of 48.7%.

Figure 5 shows a cumulative plot of the error rates for the different talkers, based on using the 10 HMM/digit set. It can be seen that only 1 of the 50 talkers had a string error rate greater than 20%. The median string error rate (UL) of 4.3%, is perhaps a better measure of how well the multi-speaker recognition system is working.

Using a similar clustering of the digit tokens of the training set, a 36 template-per-digit set was created to evaluate the performance of the template-based recognizer in the multi-speaker mode. For this system the string error rate for UL strings was 18.8%, and for KL strings it was 14.5%. These results are significantly poorer than those of the HMM based recognizer.

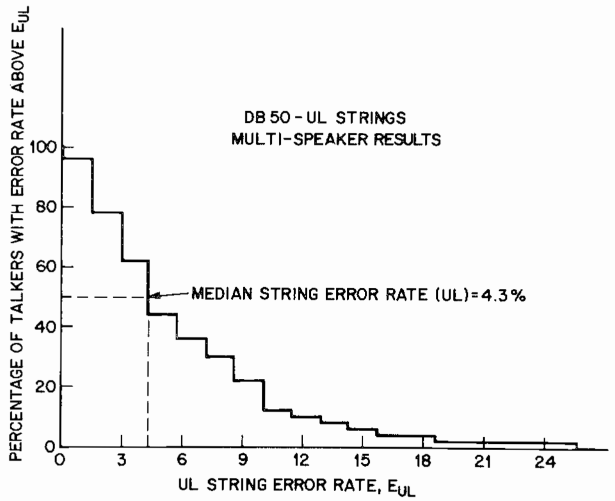

Fig. 5 Cumulative plot of the percentage of talkers with error rates above a threshold.

IV. Summary

We began this paper by talking about the importance of being able to implement a high performance connected digit recognizer. We have shown that in the case of speaker trained systems, we can achieve this goal, for essentially any talker, if an adequate amount of training is provided. Hence for 50 inexperienced, non-technical users of the recognizer, we were able to achieve greater than 98% string accuracy for unknown length strings, and greater than 99% accuracy for known length strings, over local dialed-up telephone lines. Although the amount of training used to achieve this performance is moderate (about 20 minutes of connected digit strings), it is not unreasonable for applications in which the system will be used for substantial periods on a daily basis, e.g., specialized operators, travel agents, insurance forms entry, etc. Also, in Reference 4 we showed that the amount of training could be reduced substantially (i.e., to about 2-3 minutes) with only a small increase in string error rate.

REFERENCES

[1] H. Sakoe, "Two Level DP-Matching — A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* Vol. ASSP-27, No. 6, pp. 588-595, Dec. 1979.

[2] C. S. Myers and L. R. Rabiner, "Connected Digit Recognition Using a Level Building DTW Algorithm," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* Vol. ASSP-29, No. 3, pp. 351-363, June 1981.

[3] J. S. Bridle, M. D. Brown, and R. M. Chamberlain, "An Algorithm for Connected Word Recognition," *Automatic Speech Analysis and Recognition,* J. P. Haton, Ed., pp. 191-204, 1982.

[4] 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.

[5] M. A. Bush and G. E. Kopec, "Network-Based Connected Digit Recognition," submitted for publication.

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