# A Directory Listing Retrieval System Based on Connected Letter Recognition

*L. R. Rabiner*
*J. G. Wilpon*
*S. G. Terrace*

Bell Laboratories
Murray Hill, New Jersey 07974

*ABSTRACT*

Automatic speech recognition has advanced to the stage where it is now possible to recognize connected strings of words (e.g. digits, letters, city names, airline terms) from a word reference set of isolated tokens of each of the words in the vocabulary. Recently an improved training technique, called embedded word training, was proposed in which reference word patterns were extracted from within connected word sequences themselves. In this investigation we extend the embedded word training procedure to handle letters of the alphabet for use in a directory listing retrieval task. By performing connected letter recognition of spoken names based on letter classes (rather than specific letters themselves) we show how reliable name recognition results can be achieved using a fairly straightforward name search procedure. We have tested the recognition system on 200 randomly chosen names (chosen from an 18,210 name directory) spoken at a normal rate by 4 talkers (3 male, 1 female) in a speaker trained mode. We have found that an 8% improvement in name recognition accuracy is obtained when using embedded letter training patterns over that obtained from isolated letter patterns alone. The overall name recognition accuracy was close to 95%.

## I. Introduction

Research in the area of isolated word recognition has progressed to the state where a wide variety of practical recognition systems exist both in the laboratory and in the commercial world [1-2]. The major shortcoming of these recognition systems is the isolated word format itself, since it is highly unnatural for use in a wide variety of tasks (e.g. digit dialing, word spelling etc).

The area of connected word recognition has made great studies forward in the last few years and it has reached the point where there are several laboratory and commercial systems which attain some limited degrees of success [3-6]. The basic idea in a pattern-based approach to connected word recognition is summarized in Figure 1. Assume we are given a test pattern, **T**, which represents an unknown spoken word string, and we are given a set of $V$ reference patterns, $\{R_1, R_2, \ldots, R_v\}$, each representing some word of the vocabulary. The connected word recognition problem consists of finding the "super" reference pattern $\mathbf{R}^s$,

$$\mathbf{R}^s = R_{q(1)} \oplus R_{q(2)} \oplus \cdots \oplus R_{q(L)} \qquad (1)$$

which is the concatenation of $L$ reference patterns, $R_{q(1)}, R_{q(2)}, \ldots, R_{q(L)}$, which best matches the test string **T**, in the sense that the overall distance between **T** and $\mathbf{R}^s$ is minimum over all possible choices of $L$, $q(1), q(2), \ldots, q(L)$, where the distance is an appropriately chosen distance measure.

A number of different ways of solving the connected word recognition problem have been proposed [3-6]. Although each of these approaches differs greatly in implementation, all of them are similar in that the basic procedure for finding $\mathbf{R}^s$ is to solve a time-alignment problem between **T** and $\mathbf{R}^s$ using dynamic time warping (DTW) methods.

The level building DTW based approach to connected word recognition is illustrated in Figure 2. Shown in this figure are the warping paths for all possible length matches to the test pattern, along with the implicit word boundary markers $(e_1, e_2, \ldots, e_{L-1}, e_L)$ for the dynamic path of the $L$-word match. The level building algorithm has the property that it builds up all possible $L$-word matches one level (word in the string) at a time.

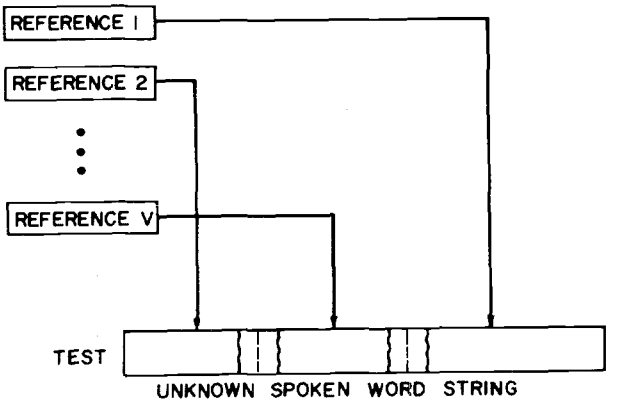


Fig. 1 Illustration of connected word recognition by concatenation of individual reference patterns.

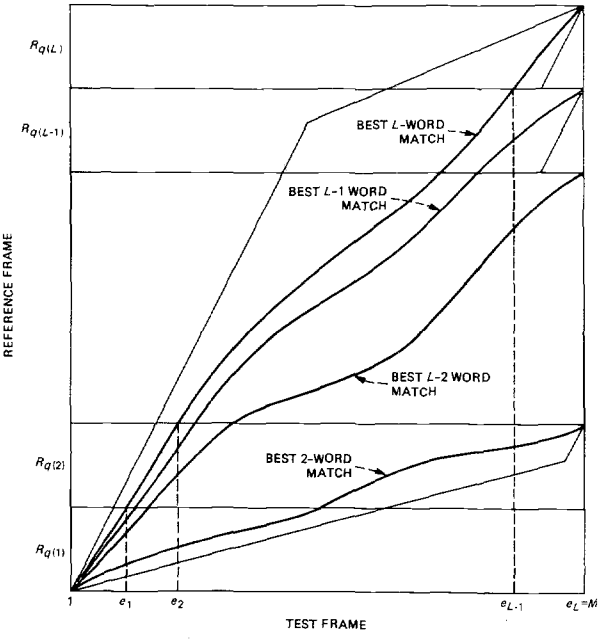


Fig. 2 Sequence of level building DTW warps to provide best word sequences of several different lengths.

Generally the single word reference patterns used in the matching procedure of Figs. 1-2 are chosen as isolated occurrences of each vocabulary word (often obtained by some form of robust training procedure [7]). This form of training is adequate as long as the rate of articulation of the spoken connected word strings is not too high (e.g. typically less than 150 words per minute). However for high rates of articulation, problems occur in the matching due to the gross differences between isolated words and those in fluent strings.

One solution to the high rate of articulation problem is to use reference tokens extracted from connected word strings to supplement the isolated word reference tokens [8,9]. Such "embedded" training tokens are extracted from known training strings and can be used in a



CH1945-5/84/0000-0385 $1.00 

modified form of the robust training procedure to give robust embedded training tokens for each word of the vocabulary.

## II. The Connected Letter Recognition System

A block diagram of the connected letter recognition system, as it used in the directory listing retrieval application, is given in Figure 3 [10,11]. A user spells the last name of the person for whom directory information is desired as a connected sequence of spoken letters, followed by a brief pause, followed by the initials (again as a connected sequence). A conventional endpoint detector finds the beginning and ending of each of the two spoken strings. An 8-pole LPC analysis is performed on each frame of both the spoken last name and the initials, where the analysis frame size is 45 msec and consecutive frames are spaced apart by 15 msec in time. For both the spoken last name, and the initials, a level building dynamic time warping (DTW) fit to a set of letter classes is made (based on letter reference patterns) and both the individual letter scores and the classes scores are saved. Last name class scores for all possible last name classes are generated and sorted by distance. A name generator sequentially goes through the sorted class list and generates all valid names within the class (i.e. those stored in the phone book). A name score generator uses the letter scores to give a total name score for each name within each class. Name scores are sorted in a list according to total name distance. Classes are searched until the best possible name score exceeds a specified threshold (related to the best name distance achieved so far). A list of the best name scores is then returned and the name recognized is the one at the top of the list.

### 2.1 Classification of Letters into Letter Classes

The concept of blocking letters into letter classes, for purposes of speech recognition, was introduced by Aldefeld et. al. [12] for the connected letter recognition application. The basic idea is that highly accurate recognition of spelled letters (over dialed-up telephone lines) cannot be achieved. Hence it is preferable to combine highly confusable letters into letter classes, perform recognition on letter classes, and decode the letter classes into actual directory names by searching a directory sorted by letter class combinations. Name scores are generated on the basis of individual letter scores (which are also generated in the recognition phase).

In particular the 26 letters of the alphabet were assigned to 3 letter classes as shown in Table I. (A fourth class, class 0, contains the space character, ϕ). Class 1 contains the /EE/ letters, whereas classes 2 and 3 are a partitioning of the remaining 17 letters into 2 disjoint sets with minimal interclass confusion. We denote the total number of classes as $C$.

For each name in the directory a set of $I$ indices for the last name are defined. These indices define the letter class of each letter of the last name. If we restrict ourselves to using $I$=6 indices for the last name, and we adopt the convention that we use the character ϕ to pad out last names of less than 6 letters, then we have a total of 1092 classes. After sorting a Bell Laboratories directory of 18,210 names, according to the letter class assignment, a total of 1053 of the 1092 classes actually had 1 or more names assigned to it. Hence coding of the last name to 6 indices is an efficient representation in terms of usage of possible letter classes.

### 2.2 Extraction of Embedded Letter Patterns

The set of letter reference patterns, for each talker, consisted of three robust tokens of each letter, obtained as follows. An isolated robust token was obtained in the conventional manner, i.e. the talker spoke the letter repeatedly until 2 tokens were sufficiently similar (small enough distance) that they could be averaged [7]. Two embedded robust tokens of each letter were obtained by having the talker speak specified 3-letter strings, extracting the middle letter via DTW alignment, and then using the standard robust training procedure on the embedded letters [8]. One of the embedded robust tokens was extracted from 3-letter strings with minimal coarticulation between letters at the boundary. The other embedded robust token was extracted from 3-letter strings with strong coarticulation between letters at the boundary. The results of the training (which typically required about 30 minutes per talker) were a set of 3 reference patterns for each letter or a total of 78 reference patterns for the 26 letters.

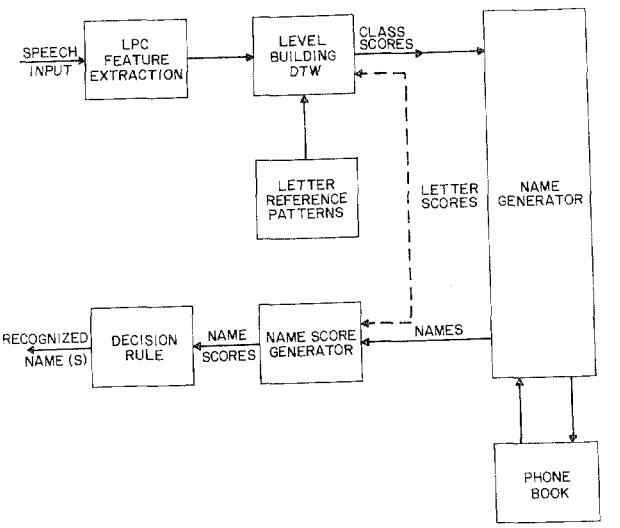


Fig. 3 Block diagram of automatic directory listing retrieval system based on connected letter spelling of names.

| Letter Class | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| ϕ | B | A | F |
| | C | H | L |
| | D | I | N |
| | E | J | Q |
| | G | K | R |
| | P | O | S |
| | T | W | U |
| | V | Y | X |
| | Z | | |

**TABLE I**
Assignment of Letters into Letter Classes

| Letter | | | | |
|---|---|---|---|---|
| A | E | S | W | |
| FAC | FEK | FSR | SWP | |
| HAP | SEQ | XSW | XWT | |
| XAT | XEK | HSL | FWK | |
| SAQ | HEK | XSN | HWQ | Non-coarticulated Sequences |
| FAP | FEQ | FSR | FWC | |
| HAQ | XEQ | HSW | SWK | |
| XAC | HEK | XSR | HWQ | |
| RAL | WEL | WSC | WWR | |
| MAN | NES | NSK | LWY | |
| PAY | MEL | LSP | MWW | Coarticulated Sequences |
| RAW | YEN | RSQ | LWR | |
| ZAL | WER | LST | MWN | |
| JAR | MEN | MSP | NWF | |
| DAN | DEY | YSK | TWN | |

**TABLE II**
Training Sequences Used for Extraction of Robust Embedded Letters for A, E, S, and W.

### 2.3 Level Building Recognition Procedure

The recognition procedure is based on using the level building (LB) DTW algorithm on strings of letter classes by using all 26 letters at each level but considering them only as different class templates. That is, different letters in the same letter class are considered as different templates for their common letter class. In the LB implementation we keep track of the $C$ best (class) candidates at each level and use the standard LB traceback algorithm [4] to generate a name class score for each of the 1053 possible last name classes.

The next step is to generate initial scores for all possible sets of 1 or 2 initials. For both the last name, and the initials, the LB algorithm keeps track of the best individual letter scores at each level. This requires a reasonable amount of storage but leads to a very efficient procedure for generating name scores. To generate a name score one merely backtracks the individual letter scores (for both last name and initials) from the appropriate memory stacks, and a total name score is generated as

$$D_{\text{NAME}} = \frac{D_{LN} \cdot L_{LN} + D_I \cdot L_I}{L_{LN} + L_I} \tag{1}$$

where $D_{LN}$ and $D_I$ are the normalized distances for the last name and initials, and $L_{LN}$ and $L_I$ are the number of letters in the last name and initials.

### 2.4 Stopping Criteria

This procedure of generating names scores is continued until a stopping criterion is satisfied. The stopping criteria is that the best possible name score for a given class exceeds the best actual name score (based on previously checked names) by a given threshold.

Once the stopping criterion was satisfied, the system returned the sorted list of name scores, and the recognized name was chosen as the one with the smallest distance.

## III. Experimental Evaluation

To evaluate the performance of the directory listing retrieval system described in Section II, 4 talkers (3 male, 1 female) each trained the recognizer using the robust training procedure to give the isolated and embedded templates for each letter. The 4 talkers were all experienced users of speech recognition systems. These same talkers each provided a test set of 50 randomly chosen names from the 18,210 name directory (the set of 50 names was different for each talker). Each name in the test set was spoken as a sequence of connected letters for the last name, followed by a pause, followed by a sequence of connected letters for the initials. The talkers spoke each name at a normal rate. All recordings were made over local dialed-up telephone lines. The average rates for the last name vary from 189 words per minute (wpm) to 218 wpm; for the initials the rates vary from 140 wpm to 167 wpm. Thus the names were spoken at very fast rates of articulation.

## IV. Recognition Results — Speaker Trained Case

The directory listing retrieval system of Section II was run on the 200 names by the 4 talkers in a speaker dependent mode. The LB parameters (see Reference [4] for a complete description of these parameters) were set to the following values:

1. $\epsilon$ = width of DTW search region = 99
2. $M_T$ = multiplier for interlevel scores = 2.2
3. $\delta_{\text{END}}$ = search region at end of string = 4
4. $\delta_{R1}$ = number of frames that can be skipped at beginning of reference template = (4,2,0)
5. $\delta_{R2}$ = number of frames that can be skipped at end of reference template = (6,3,0)
6. inserted silence at beginning or end of reference template = 2 for {b,d,g}, 3 for {p,t,k,q}.

The values for $\delta_{R1}$ and $\delta_{R2}$ were made variable with the templates — i.e. different values were used for the isolated pattern than for each of the embedded patterns. In particular, based upon preliminary experimentation, we used the set $\delta_{R1}$ = (4,2,0,), $\delta_{R2}$ = (6,3,0), where the first value is for the isolated pattern, the second value is for the non-coarticulated pattern and the third value is for the coarticulated pattern.

The inserted silence parameter is the number of frames of silence put at either the beginning or end of templates to reflect the presence of initial or final stops in the word. The letters of the alphabet for which initial silence was used were b,d,g, and p,t,k, and q. None of the letters used final silence insertion. Based on some preliminary experimentation, we used the set 2 for {b,d,g}, 3 for {p,t,k,q} where the parameter values are in terms of frames; hence 2 frames corresponds to 30 msec of silence insertion.

An analysis of the results for the chosen parameter set is given in Table III. Included in this table are the individual recognition accuracies as a function of candidate position (top $\beta$ candidates, $\beta$ = 1,2,3,4,5) for each talker (along with the average), as well as two measures of the amount of searching performed to find the best name. The search measures are $\bar{C}_s$, the average number of last name classes searched, and $\bar{N}_s$, the average number of names whose distance score was evaluated. The results in Table III show that 2 of the talkers performed well using IS templates, but the other 2 talkers performed very poorly. For these other 2 talkers the inclusion of embedded templates led to improvements in performance. It can also be seen that a 2-3% improvement in accuracy can be obtained by considering the second candidate position scores — i.e. about 2-3% of the time the correct name is in $2^{nd}$ position. Such cases are typically names with slight (within letter class) errors in the initials.

By examining the search statistics we see that the average search time for the 3 template per word set is about one-half that of the IS template set. Hence the embedded templates yield considerably more accurate name classes than that obtained from the IS templates alone.

### 4.1 Recognition Results — Speaker Independent Case

The same set of 200 names was used as a test of the connected letter directory listing retrieval system in a speaker-independent mode. For this test the letter reference patterns were a set of 12 isolated templates per letter, the templates having been extracted from a clustering analysis of isolated occurrences of each letter by 100 talkers (50 male, 50 female).

The results of the recognition test are given in Table IV which gives recognition accuracy, as a function of candidate position, for each of the 4 talkers (as well as the average), and the average search statistics. Overall it can be seen that degraded performance results from the use of only isolated templates. An average name recognition accuracy of 77% is achieved, as opposed to the 95% accuracy in the speaker dependent case. The inherent system difficulties are illustrated in the average search statistics which show that it took about 124 class evaluations, and 2179 name evaluations, to find the best name — a factor of 3 to 1 greater than required for the speaker trained case.

## V. Discussion

To get some perspective on the relevance of the results given in Section IV, we must compare the current system performance against that achieved in earlier implementations subject to the experimental constraints of small sample populations (i.e. the use of only 4 test talkers). In the most relevant comparison, Myers and Rabiner [11]

| Talker | Candidate Position 1 | 2 | 3 | 4 | 5 | $\bar{C}_s$ | $\bar{N}_s$ |
|---|---|---|---|---|---|---|---|
| 1 | 76 | 82 | 84 | 84 | 86 | 114 | 1976 |
| 2 | 80 | 82 | 82 | 86 | 86 | 139 | 2553 |
| 3 | 94 | 96 | 96 | 96 | 96 | 36 | 671 |
| 4 | 98 | 98 | 98 | 100 | 100 | 37 | 665 |
| Average | 87 | 89.5 | 90 | 91.5 | 92 | 81.5 | 1466 |

(a) Results Using IS Template Set

| Talker | Candidate Position 1 | 2 | 3 | 4 | 5 | $\bar{C}_s$ | $\bar{N}_s$ |
|---|---|---|---|---|---|---|---|
| 1 | 82 | 88 | 88 | 88 | 88 | 111 | 2086 |
| 2 | 94 | 96 | 98 | 98 | 98 | 113 | 2308 |
| 3 | 94 | 98 | 98 | 98 | 98 | 10 | 203 |
| 4 | 94 | 94 | 94 | 96 | 96 | 40 | 675 |
| Average | 91 | 94 | 94.5 | 95 | 95 | 68.5 | 1318 |

(b) Results Using IS ⊕ NC Template Set

| Talker | Candidate Position 1 | 2 | 3 | 4 | 5 | $\bar{C}_s$ | $\bar{N}_s$ |
|---|---|---|---|---|---|---|---|
| 1 | 94 | 98 | 98 | 98 | 98 | 56 | 1011 |
| 2 | 92 | 94 | 94 | 94 | 94 | 70 | 1416 |
| 3 | 96 | 98 | 98 | 98 | 98 | 19 | 352 |
| 4 | 98 | 98 | 98 | 100 | 100 | 16 | 296 |
| Average | 95 | 97 | 97 | 97.5 | 97.5 | 40.3 | 769 |

(c) Results Using IS ⊕ NC ⊕ CO Template Set

**TABLE III**

Individual Recognition Accuracies (%) as a Function of Candidate Position, and Search Statistics

| Talker | Candidate Position 1 | 2 | 3 | 4 | 5 | $\bar{C}_s$ | $\bar{N}_s$ |
|---|---|---|---|---|---|---|---|
| 1 | 88 | 90 | 94 | 94 | 96 | 29 | 473 |
| 2 | 70 | 80 | 82 | 88 | 88 | 185 | 3137 |
| 3 | 84 | 90 | 90 | 92 | 92 | 110 | 2097 |
| 4 | 66 | 74 | 74 | 76 | 78 | 174 | 3011 |
| Average | 77 | 83.5 | 84 | 87 | 88 | 124.5 | 2179 |

**TABLE IV**

Individual Recognition Accuracies (%) as a Function of Candidate Position, and Search Statistics

studied a similar system in which a fixed set of 50 names were used as the test by each of 4 talkers (different from those used here). Using isolated templates alone, recognition accuracies of 90.5% and 87.5% were achieved in the speaker dependent and speaker independent modes, respectively, for normally spoken names. Earlier work by Aldefeld et. al. [13] has shown that the 50 chosen names tended to give an overbound on true system performance as there was not a good enough representation of the common problems in name spelling (e.g. multiple names with differing initials, names differing in a single letter etc). Hence the recognition accuracies of 87% and 77% for the case of isolated templates in the speaker dependent and speaker independent modes are comparable to those reported earlier.

The recognition performance using embedded training patterns in the speaker dependent case indicates an improvement in recognition accuracy along with a significant reduction in search time to find the best name. Hence we conclude that, as in the connected digits recognition task, the use of embedded word training can and does enhance the recognition system performance.

## VI. Summary

We have shown how a practical directory listing retrieval system could be implemented on the basis of connected letter name spelling. Our results indicate that improved recognition performance can be obtained when combining embedded letter patterns (suitably extracted from 3-letter strings) with the standard isolated letter patterns to form an enhanced letter reference set. Using this combined set of references, improvements in name accuracy of 8% and reductions in search time of a factor of 2 result, when tested in a speaker trained mode using dialed-up telephone line recordings.

## References


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

[2] W. A. Lea, "Selecting the Best Speech Recognizer for the Job," *Speech Technology,* Vol. 1, No. 4, pp. 10-29, Jan./Feb. 1983.

[3] 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, pp. 588-595, Dec. 1979.

[4] 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.

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

[6] J. L. Gauvain and J. Mariani, "A Method for Connected Word Recognition and Word Spotting on a Microprocessor," *Proc. 1982 ICASSP,* pp. 891-894, May 1982.

[7] L. R. Rabiner and J. G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," *J. Acoust. Soc. Amer.,* Vol. 68, No. 5, pp. 1271-1276, Nov. 1980.

[8] L. R. Rabiner, A. Bergh, and J. G. Wilpon, "An Improved Training Procedure for Connected-Digit Recognition," *Bell System Tech. J.,* Vol. 61, pp. 981-1001, pp. 157-167, Feb. 1983.

[9] L. R. Rabiner, J. G. Wilpon, A. M. Quinn, and S. G. Terrace, "On the Application of Embedded Digit Training to Speaker Independent, Connected Digit Recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* (submitted for publication).

[10] A. E. Rosenberg and C. E. Schmidt, "Automatic Recognition of Spoken Spelled Names for Obtaining Directory Listing," *Bell System Tech. J.,* Vol. 58, No. 8, pp. 1797-1823, Oct. 1979.

[11] C. S. Myers and L. R. Rabiner, "An Automated Directory Listing Retrieval System Based on Recognition of Connected Letter Strings," *J. Acoust. Soc. Am.,* Vol. 71, No. 3, pp. 716-727, March 1982.

[12] B. Aldefeld, S. E. Levinson, and T. G. Szymanski, "A Minimum Distance Search Technique and Its Application to Automatic Directory Assistance," *Bell System Tech. J.,* Vol. 59, pp. 1343-1356, October 1980.

[13] B. Aldefeld, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "Automated Directory Listing Retrieval System Based on Isolated Word Recognition," *Proc. IEEE,* Vol. 68, pp. 1364-1379, November 1980.