ADAPTATION OF REFERENCE PATTERNS IN WORD-BASED SPEECH RECOGNITION

Fergus R. McInnes

Thesis submitted for the degree of PhD
University of Edinburgh
1988

ABSTRACT

The word-based approach to automatic speech recognition is one which has received attention from many researchers and has been exploited in various practical applications. A typical recognition system has a set of stored reference patterns, one or more for each word in the vocabulary to be recognised. These reference patterns are formed from training utterances supplied before a recognition session begins, either by the intended user of the system or, for a speaker-independent system, by a representative set of speakers. When the system is used for recognition, each new input utterance is compared with the stored patterns and is recognised as the word (or sequence of words) for which the minimal value of a distance (dissimilarity) measure, or equivalently the maximal likelihood, is obtained. The comparison of the input with the reference patterns is typically accomplished by an algorithm incorporating dynamic programming, which finds the optimal alignment of input and reference patterns and the corresponding distance or likelihood.

This approach to recognition, in its basic form, retains the same reference patterns unchanged throughout the recognition of any sequence of input utterances. Thus the recognition system has no capability of learning from the new utterances presented during a recognition session. If a recognition system can be made to adapt its reference patterns during its operation, to incorporate information from the recognised utterances, then this may be expected to allow progressive improvement of the modelling of the words (as pronounced by the current speaker), and hence enhancement of the accuracy of recognition – provided that the adaptation of incorrect words' reference patterns in cases of misrecognition can be prevented or kept to a sufficiently low level. By adaptation, speaker-specific initial reference patterns can be made more reliably representative of the speaker's typical pronunciations, by the use of data from additional utterances of the words; and speaker-independent reference patterns can be made speaker-specific through the incorporation of information from utterances by the speaker currently using the recognition system. Adaptation can also permit the dynamic adjustment of reference patterns to track any gradual drift, or systematic difference from one occasion to another, in the speaker's voice or pronunciations or in the level and characteristics of background noise.

In this thesis, the development of an isolated word recognition system which incorporates various adaptation options is described, and the results of experiments to measure the effects of adaptation are presented and discussed. Both supervised adaptation (which is controlled by feedback from the user as to the correctness or incorrectness of each recognition) and unsupervised adaptation (without such feedback) are explored. The adaptation operates by a weighted averaging of the current reference pattern (template) with the recognised input. Two main weighting options have been defined: one which results in optimisation of the templates for the speaker's typical realisations of the words (if these are assumed to be invariant in time), and one which results in tracking of gradual variations in time. Various values of the relative weights on the existing template and on the input have been tested. Adaptation has been applied both to speaker-specific initial templates and to speaker-independent ones. In each case, the statistical significances of comparative results are computed from the means and variations across a set of test speakers.

A compensation technique has been introduced, whereby the distance obtained in matching a template with an input utterance is adjusted according to the number of times that template has been adapted. This is necessary because adaptation reduces the typical distances obtained for the adapted template even when this template does not correspond to the correct recognition of the input. Appropriate values of the compensation parameters, to optimise the recognition performance, have been found for various adaptation options.

The main conclusions from the experiments are that adaptation, especially supervised adaptation, can yield consistent and useful improvements in the performance of an isolated word recognition system, and that the application of appropriate word distance compensation is important for the attainment of the maximum benefit from the adaptation.

Possible refinements and extensions of the adaptation technique are discussed. Results of a limited evaluation of template adaptation in a connected word recognition system are presented.

Other aspects of the recognition system which are described and discussed include an efficient multiple-stage decision procedure and some features of the user-system interface design.

TABLE OF CONTENTS

DETAILS OF PUBLISHED PAPERS APPENDED

Paper 1 [97]:

F.R. McInnes, M.A. Jack and J. Laver, "Comparative study of time segmentation and segment representation techniques in a DTW-based word recogniser" – reproduced from IEE Conference Publication No.258 (proceedings of the IEE International Conference on "Speech Input/Output; Techniques and Applications", London, 24-26 March 1986), pp.21-26, by permission of the Institution of Electrical Engineers.

Paper 2 [255]:

F.R. McInnes, M.A. Jack and J. Laver, "An Isolated Word Recognition System with Progressive Adaptation of Templates" – reproduced from Proc. Inst. of Acoust., vol.8, part 7, pp.283-290, 1986 (proceedings of the Institute of Acoustics Autumn Conference, Windermere, 28-30 November 1986).

Paper 3 [258]:

F.R. McInnes, M.A. Jack and J. Laver, "Experiments with Template Adaptation in an Isolated Word Recognition System" – reproduced from Proceedings of the European Conference on Speech Technology, Edinburgh, 2-4 September 1987, vol.2, pp.484-487, by permission of CEP Consultants Ltd.

Paper 4 [259]:

F.R. McInnes and M.A. Jack, "Reference template adaptation in speaker-independent isolated word speech recognition" – reproduced from Electronics Letters, vol.23, no.24, 19 November 1987, pp.1304-1305, by permission of the Institution of Electrical Engineers.

LIST OF ILLUSTRATIONS

LIST OF TABLES

ACKNOWLEDGEMENTS

I acknowledge with thanks the contributions of all those who have helped to make this thesis possible – notably the following:-

Professor Mervyn Jack, who guided me into the field of template-based speech recognition; who as my supervisor has encouraged me in pursuing this research and in publishing the results; and who also read through much of the text of this thesis during its composition and suggested improvements;

the Science and Engineering Research Council, which has supported me financially through a research studentship;

Professor John Laver, the Director of the Centre for Speech Technology Research, whose initiative in setting up the Centre has resulted in a suitably equipped and congenial environment in which to explore automatic speech recognition;

those who have lent their voices to contribute speech data for my experiments – in particular Jocelyn Trehern and Anne Johnstone (speakers 2 and 3 in the data base of section 4.2.1), and Edmund Rooney, Pamela Rodriguez and Shona Anderson (speakers 2, 3 and 4 in the data base of section 6.2);

George Fletcher, who translated the original Fortran recognition program into C, provided the digitisation component for *del*, and improved the endpoint detection procedure; and George Duncan, who composed the LPC analysis software;

many other colleagues in the Electrical Engineering Department and in the Centre for Speech Technology Research, who have given helpful advice, cooperation and information, especially with regard to computer facilities and recording equipment;

colleagues elsewhere who have stimulated my research through their comments and discussions at conferences and seminars, and particularly John Bridle of the Royal Signals and Radar Establishment who has shown an interest in my work;

my father, Bennet McInnes, who is also a research scientist, and with whom I have had some stimulating and enlightening discussions as to the nature of research work;

my mother, Elizabeth McInnes, who has passed on to me the enthusiasm

for language which contributed to my entering this field of research, and who has been supportive throughout the project;

the other friends who have taken an interest in my progress, and in particular those in the West Pilton Growth Group of Granton Baptist Church, who prayed for me during the final preparation of this thesis – and God who has answered their prayers and enabled me to complete it.

While others have contributed to this thesis in many ways, I accept responsibility for the text of the thesis and the research described in it.

DECLARATION

This thesis has been composed by me and describes original work of my own execution. I composed all the computer programs used in the research and described in the thesis – with the exceptions noted in the acknowledgements on page xv.

CHAPTER 1

INTRODUCTION

1: INTRODUCTION

The field of automatic speech recognition is one which has received attention in recent years among researchers in engineering, computer science, artificial intelligence, phonetics and linguistics. There is a wide range of potential applications for machines which can recognise and respond to spoken words, phrases or sentences.

(Examples of actual and potential applications include entry of data or instructions to computers and automated systems when the user's hands or eyes are busy for some other task, making the use of a keyboard inconvenient – as for instance in parcel handling, quality control inspection, computer-aided design, air traffic control and cartography [208,209,211,213,214,215,216,218,219,220]; telephone information and transaction services [16,128,129,221,223,227]; and control of computers and other equipment by those who are physically handicapped and so cannot use a keyboard [96,208,210,212,220,223]. Voice input may also be advantageous in physically hostile environments [215,216,217,238,239], since a microphone is easier than a keyboard to protect from dirt, harsh weather or vandalism. If reliable recognition of fluently spoken sentences of unrestricted natural language can be achieved, this will open up many more applications, such as text composition by dictation to a machine; but this goal is an ambitious one and as yet unattained [15,216,220].)

Various technical approaches have been developed, some of them based on the identification of (phoneme-sized) phonetic segments in the speech to be recognised [15,171,174,175,178], and others based on the recognition of larger units such as diphones [180], demisyllables [173], syllables [248] and whole words [12]. In each case, the main source of difficulty is the inherent variability of speech,

whereby realisations of the same phoneme, syllable, word or other unit vary widely in their acoustic characteristics from one occasion or context to another. The acoustic signal corresponding to any linguistic unit will depend on the immediate and wider context in which it is pronounced; on various characteristics of the speaker, some of which vary significantly in time; and on features of the physical environment including the level and type of background noise. Many of these factors cannot easily be controlled or predicted [15].

Speech recognition strategies using units smaller than the word, especially those which work by identifying phonetic segments, must cope with a high degree of variation in the pronunciation of each such unit resulting from its interaction with the preceding and following speech [15]. Recognition systems based on such phonetic units generally have to include a large amount of linguistic knowledge, expressed for instance in a set of phonological rules [9,11,15], to represent the possible variations of each unit in different phonetic environments.

Approaches based on recognition of whole words, without their segmentation into smaller phonetic or linguistic units, can avoid much of the problem of context-dependent variability, since the immediate phonetic context of each part of a word is similar on each occasion when the word is uttered. In particular, in the isolated word recognition task (a limited but useful case of speech recognition), where the input consists of single words spoken with pauses between them, there is very little effect on whole-word patterns due to immediate context – although in a sequence of isolated words variations may occur, especially in intonation, according to the position of each word in the sequence. In the isolated word case, each utterance to be recognised consists of just one unit, and can be recognised directly by comparison with reference patterns which correspond to the words in the system vocabulary. In the more complicated case

of connected speech (with careful enunciation), the inter-word coarticulation and assimilation effects which occur [15] affect mainly the beginnings and ends of the words.

This whole-word-based pattern-matching approach to speech recognition is a simple one from a linguistic point of view, in that it makes very little use of knowledge about the structure and characteristics of spoken language. The pattern-matching algorithms used are not specific to speech, and may be used for any of a wide variety of problems involving the recognition of sequential patterns, such as handwritten character recognition. However, the simplicity of the approach and the ease of its implementation (by comparison with those methods using phonetic units and detailed speech knowledge) have allowed it to be exploited in commercially viable products, while the more sophisticated knowledge-based approaches have remained mostly in the research stage.

One form of variability that affects word-based speech recognition is the non-linear extension and compression of the timescale of a word from one utterance of it to another. Early isolated word recognition systems based on whole-word reference patterns [4] used linear time-alignment of input and reference words, which was not always satisfactory. A major advance in the development of successful isolated and connected word recognition was the introduction of non-linear time-alignment techniques using dynamic programming [1,2,9,18,19].

This thesis is concerned with certain developments of the word-based speech recognition technique using dynamic programming alignment. In particular the thesis highlights the improvement in system performance obtainable by adaptation of the reference patterns to the recognised input words. The results presented are mainly for isolated word recognition, but the extension to connected word recognition will be considered briefly.

The mathematical principles involved in the basic pattern matching technique incorporating dynamic programming alignment, and the results of experiments by other researchers with various formulations and extensions of this technique, are reviewed in chapter 2. In chapter 3, some aspects of these results are discussed, and areas of particular interest for further research are identified. The remaining chapters describe experiments conducted in order to investigate these areas, and give an analysis of the results obtained together with a discussion of their implications.

Chapter 4 describes some preliminary experiments comparing several forms of linear and acoustically-based segmentation of word reference patterns which can be used as preprocessing for a template-matching isolated word recogniser, and the development of a multiple-stage recognition system which incorporates these segmentation techniques to provide a substantial reduction in the amount of computation required to recognise each word.

Chapter 5 describes the further development of the system to incorporate an interactive recognition mode and several options in template adaptation.

Chapter 6 contains the results of experiments with adaptation of speaker-specific initial templates. The different forms of adaptation are compared, and some observations on the interaction between the system and the user are given.

Chapter 7 describes experiments carried out with speaker-independent initial templates, exploring the effectiveness of adaptation for making the templates correspond more closely to the pronunciations of a particular speaker.

Chapter 8 contains a review and summary of the main features of the results, and a discussion of possible extensions, developments and applications of the techniques explored.

An appendix describes the statistical analysis which was applied to the results of the experiments.

The references are arranged by subject category, and within each subject category by date of publication. An alphabetical index of authors is provided, following the reference list.

Several papers [97,255,258,259] containing brief statements of some of the results of the author's research are reproduced at the end of the thesis.

CHAPTER 2

REVIEW OF WORD-BASED SPEECH RECOGNITION USING DYNAMIC PROGRAMMING FOR TIME ALIGNMENT

2: REVIEW OF WORD-BASED SPEECH RECOGNITION USING DYNAMIC PROGRAMMING FOR TIME ALIGNMENT

2.1: Introduction

Most currently available automatic speech recognition systems rely on comparison of the input speech to be recognised with stored reference patterns, with each reference pattern representing one of the words of the vocabulary being used. The stored pattern for each word may be a template [16,18,19,48], derived from one or more training utterances (tokens) of the word obtained before the recognition session begins; or a statistical model of the word's characteristics, derived from multiple training utterances [16,104,119]. (In fact, the use of templates can be viewed as a special case of statistical modelling, with some simplifying assumptions of uniformity in the structure of the model which allow it to be derived from a small amount of training data. The relation between templates and a more general class of statistical models is explored in detail in section 2.5 below.) In either case, it is usually necessary to time-align a series of input vectors, each containing spectral information derived from a short section (frame) of the input speech, with the reference pattern. In this way it is possible to compute a measure of the similarity between the input word and the word which that reference pattern represents.

In many word-based speech recognition systems, the alignment of input and reference patterns is accomplished using the optimisation technique known as dynamic programming [1]. Where the reference patterns are templates, the dynamic programming alignment procedure [2,18] is known as dynamic time warping [51,60]. (The phrase "dynamic time warping" is not universally approved among speech recognition researchers [16]; however, it will be retained

in this thesis, as the most generally understood term for this particular application of dynamic programming.) The corresponding procedure for hidden Markov models [16,104,119] (a commonly used class of statistical models) is called the Viterbi algorithm [5,104].

This chapter reviews the variants and developments of the basic methods which have been devised and tested by numerous researchers, and gives a comparison of the results obtained. Much of the material in this chapter has appeared elsewhere [17].

In section 2.2, a description is given of the operation of a word recognition system using template matching; the time registration problem for a template and an unknown utterance is stated; and the basic principles of dynamic time warping (DTW) as applied to this problem are explained.

In sections 2.3-2.7, various options in the design of speech recognisers using DTW are described. Applications to isolated word recognition, spotting of key words in continuous speech and connected word recognition are considered. Methods for improving the computational efficiency of a recogniser using DTW are described. The Viterbi and forward-backward algorithms, for recognition using hidden Markov models, are introduced in section 2.5, and the conceptual unity of the DTW and Viterbi algorithms is demonstrated. In each section, the results of experiments by various researchers comparing the performances of different techniques and algorithm formulations are summarised.

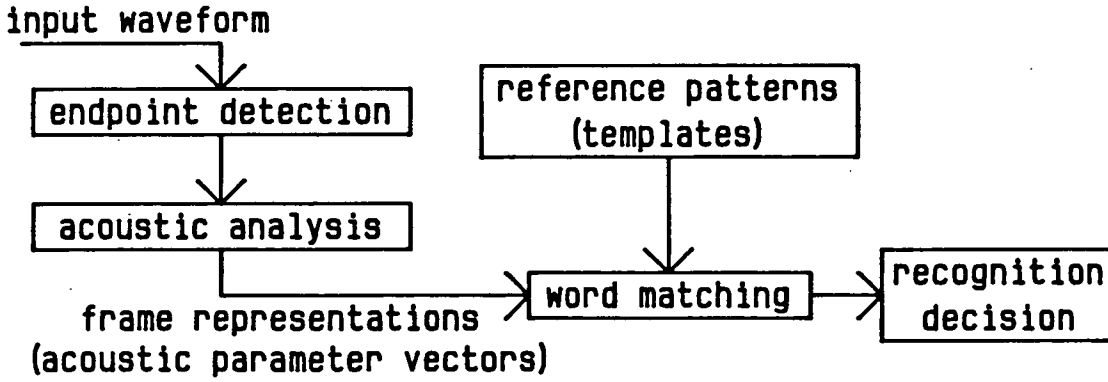
Finally, in section 2.8, a brief discussion and summary of the results is given. Some of the main points emerging from these results, which have a particular bearing on the topic of this thesis, are discussed further in chapter 3.

2.2: Template matching and time registration

The basic idea involved in template matching is that each word in the system vocabulary is represented by a template (in some cases more than one [48,54,56]), which is a reference pattern created from speech data and stored in the machine's memory. Each unknown input word to be recognised is compared with the stored templates and identified as an instance of that vocabulary word whose template best matches the unknown input.

Each word template consists of a sequence of representations of short time segments or "frames" of the reference speech waveform. The representation for each frame may be a vector of bandpass filter outputs, or a set of autocorrelation and/or linear prediction coefficients, or some other set of parameters such as cepstral coefficients. The unknown input speech waveform is similarly processed into input frame representations.

Figure 2.1: a typical isolated word recognition system

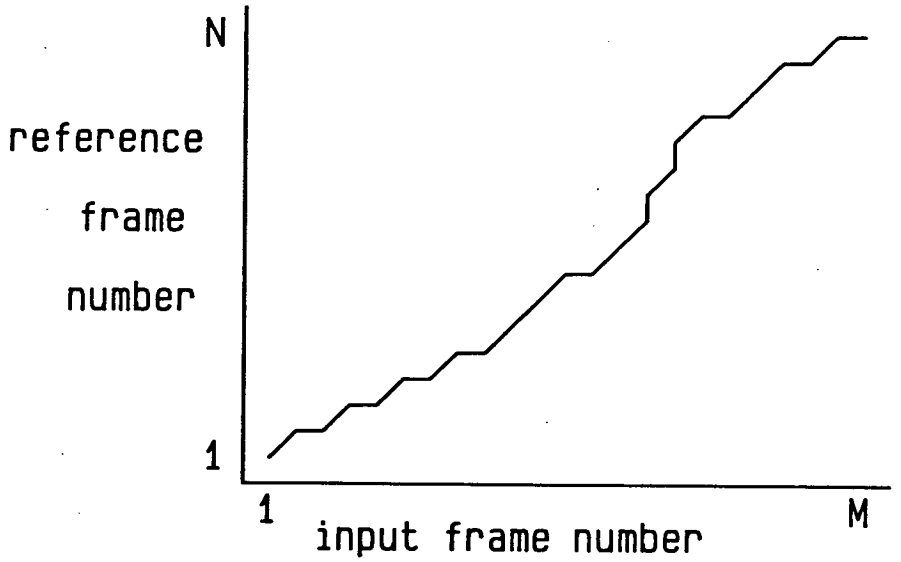


The components of a typical isolated word recognition system are shown in figure 2.1. The input waveform is digitised, the beginning and end of the word are identified and an acoustic analysis is performed in each frame of the (endpoint-detected) word. The resulting sequence of frame representations is matched against each stored word template in turn, so as to identify the word which has been spoken.

In comparing the input with a template, each input frame is matched with a reference frame from the template, and a frame distance is computed, which is a measure of how different the two frame representations are. (An alternative formulation is in terms of similarities instead of distances; the details of the algorithm are the same in either case, except that, wherever a distance is to be minimised, the corresponding similarity is to be maximised.) Then an overall distance is computed from the frame distances for all the matched pairs of frames. The sequence of matched pairs of input and reference frames forms a time registration path, which can be depicted as a graph of reference frames against input frames, as shown in figure 2.2. The point (m,n) on the path, where m and n are integers, corresponds to the matching together of the mth input frame and the nth reference frame. The slope of the path represents the degree of compression (expansion, where the slope is less than 1.0) applied to the template in aligning it with the input frames. In particular, a vertical step in the path corresponds to the matching of two successive reference frames to the same input frame, and a horizontal step corresponds to the matching of the same reference frame to two successive input frames.

The overall distance is a weighted sum of the individual frame distances for the pairs of frames on the path. The weight given to each frame distance can be made to depend on the slope of the time registration path near the point defined by the pair of frames in question.

Figure 2.2: a time registration path



The time registration problem is that of finding the best possible time registration path for given input and reference data, i.e. the path which minimises the overall distance subject to appropriate constraints. Three obvious types of constraints are the following:-

Endpoint constraints: the beginning and end of the input data must be matched with the beginning and end, respectively, of the reference data. This requirement may be relaxed to allow for inaccuracies in the identification of endpoints when the frame sequences were created.

Continuity: successive points on the time registration path should be close together (in both the input and the reference dimensions).

Monotonicity: as the path progresses in input time, it should move forward (not backward) through the reference template.

Further constraints can be imposed, such as restrictions on the slope of the path, on the sharpness of changes of slope, or on the region of the input-reference plane in which points on the path may lie.

The simplest way to construct a path satisfying the constraints is linear time registration, in which the path is made to correspond as closely as possible to a straight line joining the initial and final points. This may be achieved either by constructing an approximately straight-line path through the array of integer points (m,n) (which may not be of equal extent in the input and reference directions, since the two utterances may be of different durations) or else by normalising both patterns (input and reference) to the same length so that an exact linear path (of slope 1.0) can be defined. This linear registration is not totally satisfactory for speech recognition, especially where polysyllabic words or connected strings of words are being matched, since the speeds at which different parts of an utterance are spoken can vary independently of one another.

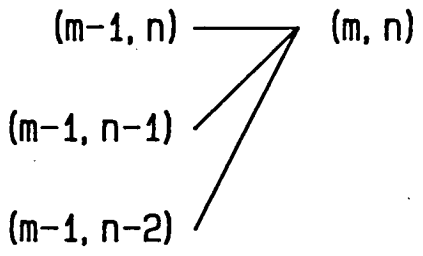
It would be possible to solve the time registration optimisation problem by computing overall input-template distances for all time registration paths satisfying the constraints and selecting the path giving the smallest overall distance. However, this would involve an excessive amount of computation.

A much more efficient procedure relies on the application of dynamic programming [1]. (In this phrase, the word "programming" is essentially an optimisation-algorithm term, not a computer-science term.) Dynamic programming is relevant to a whole class of problems where a path is to be chosen from an initial point to a final point so as to minimise or maximise a sum of quantities which correspond individually to the individual segments of the path. The essential principle involved is that, if a point, P, lies on the optimal complete path, then the partial path (i.e. the part of the optimal complete path) from the

initial point to P is the optimal path from the initial point to P. Thus, if the optimal path from the initial point to P has been found, any other path from the initial point to P can be discarded, and excluded from further consideration, in the attempt to construct the optimal complete path.

This dynamic programming procedure when applied to the time registration problem takes the form of a dynamic time warping (DTW) algorithm. The algorithm proceeds along the input one frame at a time and, for each successive input frame, computes a frame distance $d(m,n)$ and an accumulated distance $D(m,n)$ for each value of n permitted by the search area constraints (where m is the input frame number and n is the reference frame number). The accumulated distance is the weighted sum of the frame distances on the optimal partial path from the initial point to (m,n) and is found by optimising over the points permitted as predecessors of (m,n) on such partial paths. Thus, in the simple example illustrated in figure 2.3, where the constraints permit $(m-1,n-2)$, $(m-1,n-1)$ and $(m-1,n)$ as previous points, and the weighting on each frame distance is 1.0 (i.e. the accumulated distance is simply the sum of the

Figure 2.3: simple local path constraints



frame distances on the path so far),

$$D(1,1) = d(1,1) \tag{2.1}$$

(and D(1,n) is treated as infinite for $n > 1$), and for $m > 1$

$$D(m,n) = \min \{D(m-1,n-2),D(m-1,n-1),D(m-1,n)\} + d(m,n). \tag{2.2}$$

When the final point (M,N) is reached, the overall distance between the input and the template (with optimal time alignment) is simply $D(M,N)$.

2.3: Options in a DTW system for isolated word recognition

The simplest application of DTW for speech recognition is to the recognition of a single word spoken in isolation, as already described in section 2.2. The ideas introduced in this section apply primarily to this isolated word recognition problem, but will also be applicable (with appropriate modifications) to the more complex problem of recognising words in connected speech.

2.3.1: Frame representations and distance measures

The first requirement for a template matching system is that there should be a method of representing the speech waveform in each time frame, with an associated distance (or similarity) measure for comparing frame representations. Various types of representation have been reported, notably those based on bandpass filtering [19,23,28,30,31] and those based on linear predictive coding (LPC) analysis [18,19,20,22,23,24,36,47].

2.3.1.1: Bandpass filter representations

A bandpass filter system [31] has a filter (analogue or digital) for each of a number of frequency bands covering the range of frequencies being used. For each time frame, a measure of the signal energy within each filter band is computed. The filter outputs (energy levels) are usually transformed logarithmically and then normalised in each frame by subtracting the overall log energy in that frame. Filter outputs can be used either directly as frame representations or to compute some other representation, such as cepstral coefficients [23], which provide a convenient expression of the overall (smoothed) shape of the speech spectrum. These representations can also be computed from discrete Fourier transform (DFT) coefficients [23].

The normalisation of each frame vector by subtracting the overall log energy in the frame entails the loss of information in that all parts of a word are adjusted to the same loudness. This can be corrected by including in the frame representation a measure of the overall energy in the frame, or by some normalisation procedure taking into account the energy level of the utterance as a whole [58,76].

The number of bandpass filters used has varied considerably from one system to another, but has usually been in the range 6-20. Increasing the number of filters used generally improves the rate of correct recognition achieved by the system [19,31], although it has been observed [31] that recognition rates for female speakers decline when the filter bandwidths become too narrow. The frequency spacing of the filters can be linear [31] (with a constant frequency difference between adjacent filter bands), logarithmic [19,31] (e.g. octave or third-octave filters) or some other non-linear spacing – such as the mel frequency arrangement [23], in which the filters are linearly spaced up to about

1000Hz, and logarithmically spaced over higher frequencies, to accord with the distribution of significant information in speech. Cepstral coefficients derived from filters on a mel frequency scale have been found to yield good recognition performance [23,72,146,160]. The filter bands can be non-overlapping or overlapping [31].

The distance between two frame representations consisting of filter energies or cepstral coefficients is often taken to be the absolute value (L1) norm of the difference of the two vectors, i.e.

$$d_{\mathrm{ABS}}(m,n) = \sum_{k=1}^{Q} |X(m,k) - Y(n,k)|, \tag{2.3}$$

where $X(m,k)$ is the kth component of the mth (Q-dimensional) input frame representation and $Y(n,k)$ is the kth component of the nth reference frame representation [19]. (This absolute value distance measure is also called the city-block distance [32], or sometimes the Chebyshev norm [19,49]. However, this use of the term "Chebyshev norm" or "Chebyshev metric" is best avoided, since historically the name of Chebyshev is associated with a norm consisting of the maximum – rather than the integral (for functions) or sum (for vectors) – of absolute values [163].) The Euclidean norm of the vector difference,

$$d_{\mathrm{EUC}}(m,n) = \left[\sum_{k=1}^{Q} (X(m,k) - Y(n,k))^2 \right]^{\frac{1}{2}}, \tag{2.4}$$

or its square,

$$d_{\mathrm{SQ}}(m,n) = \sum_{k=1}^{Q} (X(m,k) - Y(n,k))^2, \tag{2.5}$$

can be used as a distance measure [23], but this requires more computation than the absolute value distance [19]. Depending on the other characteristics of the system, the Euclidean metric may give recognition results slightly better than those obtained with the absolute value metric [19,58], or in some cases slightly worse [30]. These two distance measures are the most widely used, but various others have been devised [26,27,30,36,66]. In particular, for filter energies or formant-based representations, distance measures using dynamic programming for non-linear alignment in the frequency domain have been found to yield enhanced recognition rates, with a considerably increased amount of computation [37,43]; and, in the case of a cepstral representation, applying different weights to the individual coefficients before the distance computation has been observed to improve the performance [39,40]. If a mixed set of acoustic parameters is used as the frame representation – if, for instance, an overall energy term is included – then it will usually be necessary to assign different weights to the different types of components in the distance measure [36,38,42].

If the system is to operate in noisy conditions and the spectrum of the noise is known or can be measured, a noise masking or compensation technique can be applied [35], which will result in modified filter energy vectors or a modified frame distance function. Such a technique can greatly improve performance at low signal-to-noise ratios.

2.3.1.2: LPC representations

The LPC approach [18,20,21,22,47] involves sampling the (time domain) speech waveform (typical sampling rates being from 6 to 20kHz) and then estimating prediction coefficients $(a(1),a(2),...,a(p))$ for the sequence of sample values $(...,s(t-1),s(t),s(t+1),...)$ so that the mean squared value of the error term

$e(t)$ defined by

$$e(t) = s(t) + \sum_{i=1}^{p} a(i)s(t-i) \tag{2.6}$$

is minimised. The mean squared error (σ^2) is called the prediction residual, and is given by

$$\sigma^2 = \mathbf{a}R\,\mathbf{a}', \tag{2.7}$$

where **a** is the row vector $(1,a(1),...,a(p))$, "'" denotes "transpose" and R is the $(p+1)\times(p+1)$ autocorrelation or covariance matrix of the signal samples. Before the prediction coefficients are calculated, preemphasis [75] is usually applied to the sequence of sample values; this reduces the energy at low frequencies, thus compensating for the spectral tilt which is typical of voiced sounds in speech, and can result in an improvement in the performance of the recogniser [22]. (Preemphasis may be applied, also, as preprocessing for a filter bank [19,30].)

The order of the LPC analysis, p, has varied from system to system. The value of p required for adequate modelling of the vocal tract depends on the sampling frequency for digitisation of the signal: the higher the sampling frequency, the larger the analysis order p should be. It has been suggested that, when the sampling frequency in kHz is n, the analysis order should be at least $n+4$ [20]. Thus values in the region of 10 may be adequate for telephone speech, which has a bandwidth of about 4kHz and is typically sampled at between 6 and 7 kHz [48,87]; where a wider-band signal is available, and a higher sampling frequency is employed to make use of this, the analysis order should be increased. In practice, 8th-order analysis has often been used for tele-

phone speech [36,48,60,90], while for speech sampled at 10kHz analysis orders from 10 to 14 have been applied [19,23,47,134,160], and the recognition results obtained with 10th-order analysis have been observed to be as good as those with 14th-order [47].

The windows of sample values for consecutive frames are overlapped, to produce typically 45ms windows at 15ms separation, with 30ms overlap between adjacent windows [12,18,36,60] (although shorter windows have also been applied [23,48,75]). Two methods for estimating prediction coefficients, the autocorrelation and covariance methods, are compared in [22]. In estimating the vector **a** by the autocorrelation method, a windowing function is used – one frequently used example being the Hamming window [12,22].

A possible distance measure between LPC representations is the Itakura metric or "log likelihood ratio" [18,22], defined by

$$d(m,n) = \log \frac{\mathbf{a}_{ref}(n) R_{in}(m) \mathbf{a}_{ref}(n)'}{\mathbf{a}_{in}(m) R_{in}(m) \mathbf{a}_{in}(m)'}, \tag{2.8}$$

where $\mathbf{a}_{in}(m)$ and $\mathbf{a}_{ref}(n)$ are the vectors of prediction coefficients for the input and reference frames and $R_{in}(m)$ is the autocorrelation or covariance matrix for the input frame. This is the log ratio of the prediction residuals when the input signal is predicted using the coefficients derived from the reference data and when it is predicted using the (optimal) coefficients derived from the input itself. (The word "metric" is used loosely in describing this function, since it does not satisfy the requirements of the mathematical definition of a metric; in particular, it is not symmetric, i.e. it does not necessarily take the same value if the reference and input data are swapped.) Other LPC distance measures, involving the prediction coefficients and autocorrelations (or covariances) of the reference and input samples, have been devised [21,24,27,36]. Some of these, which are

symmetric with respect to the two frames being compared, have been found to have better properties for distinguishing correct and incorrect frame matches than the Itakura metric [24].

The performance of LPC-based recognisers in noisy conditions can be improved by filtering the noisy signal before carrying out the LPC analysis [29,134].

Other representations such as reflection coefficients and cepstral coefficients can be derived from an LPC analysis [21,23,134], and for these the absolute value and Euclidean distances described above can be used. A theoretical and experimental study of several LPC-based distance measures is given in [21]: it is shown that the Euclidean distance on cepstral coefficients and a symmetric likelihood-ratio ("cosh") distance are approximations to the root-mean-square (r.m.s.) distance between log spectra.

The autocorrelation coefficients can be used directly as a speech representation, rather than to compute prediction coefficients; a distance measure allowing for autocorrelation lag offsets between the two frame representations compared has been formulated [235].

2.3.1.3: Comparison of representations

Comparative studies of LPC and bandpass filter representations [19,31] suggest that an LPC representation gives better results on telephone-quality speech, which is band-limited so that high frequencies are lost, but not on speech without this band-limiting.

In a comparison of several LPC-based and filterbank-based representations for recognition of monosyllabic words [23], the best recognition accuracy was obtained using cepstral coefficients derived from filters on a mel frequency scale.

Among the LPC representations, the prediction coefficients with the Itakura distance yielded the best results (only a little poorer than those with the mel cepstral representation), and cepstral coefficients were better than reflection coefficients.

Speech representations based on models of human auditory processing have been proposed [33,34,41]. In some cases these have been found to yield improvements in recognition performance over the more conventional representations [41].

In experiments into the representation of the dynamic characteristics of speech [42], better recognition results were obtained by the use of a linear combination of each cepstral coefficient and its time derivative (estimated by regression analysis) than with either the instantaneous value or the derivative alone. A similar improvement was achieved by using the cepstral coefficients and their time derivatives as separate parameters [38]; but the linear combination technique has the advantage that it does not increase the dimension of the frame representations, and so does not add to the computation for each frame distance. Some further improvement was obtained by using the log-energy derivative as an extra component of each frame representation. It has also been shown [44,45] that improved recognition performance can be obtained by using a concatenation of the feature vectors from two frames with a 40ms time separation and applying a transformation to adjust the distance for inter-frame covariance. (The recognition accuracy was increased even more when the transformations were made specific to reference frames, and when components of the vectors were selected to optimise the discrimination of different words; but this was possible only by the use of a large number of training utterances, and these techniques also increased the computational load significantly.)

2.3.2: Training and template creation

A speaker-trained (speaker-dependent) word recogniser, in which the templates are created from speech data from the specific speaker who will speak the input words to be recognised, can operate using just one template, derived from a single reference utterance, for each word in its vocabulary. The speaker has to utter a full list of the words in the vocabulary to train the system (i.e. provide it with reference data) before using it to recognise subsequent speech input. It has sometimes been found beneficial to have several templates for each word [56,60,76,151], or to derive each template from two or more utterances of the word [23,56,76,86,151]; in either of these cases, the speaker must provide more than one utterance of each word for training.

This requirement of training to each new speaker is acceptable if the vocabulary is small or if only one speaker is to use the recogniser. However, for a system with a larger vocabulary and a high turnover of different speakers, it becomes a limitation. There are at least two possible options for avoiding the training requirements of this fully speaker-dependent mode of operation. These are (a) the speaker-adaptive mode, in which the recogniser is trained to each new speaker using a selected subset of the full vocabulary (from which speaker-adapted templates are deduced for the remaining words using previously stored speaker-independent information); and (b) the fully speaker-independent mode [48,54,55], in which templates are created from the speech of a limited number of training speakers, and are then used in the recognition of words spoken by whatever speakers may use the system. (The speaker-adaptive mode of training referred to here is different from the adaptation which will be considered in detail in chapters 6 and 7 of this thesis, in that it requires at least a short training session to derive the adapted templates before the recognition can com-

mence, and after this session no further adaptation is performed.)

The first of these possibilities is difficult because of the complexity of the pattern representing a word. Using any of the types of frame representation described above, large amounts of previously stored information and computation are likely to be required to adapt every frame of every vocabulary word to the characteristics of a new speaker (as determined from the smaller training vocabulary). A system using pretraining and a statistical analysis of the data from the pretraining speakers has been reported [237], but this uses phoneme templates rather than word templates. Other adaptive recognition systems that have been developed or proposed [11,172,242,243,248,249,251] also rely on having reference patterns (or synthesis parameters [242]) for phonetic or syllabic units, from which the word reference patterns are built. However, adaptation to word-independent speaker characteristics by a spectral transformation is feasible in a word-based recogniser and has been applied with some success [240,252]. Speaker adaptation methods based on vector quantisation (see section 2.4 below) have also been devised [244,250,257].

The second possibility, a speaker-independent recogniser, has been investigated by various researchers [26,47,48,53,54,55,89,90,99,100,102]. It is here that multiple reference utterances for each word in the vocabulary become very important. Because of the variations in pronunciation among speakers, a template derived from an utterance of a word by a single training speaker may not be adequate for recognition of the same word spoken by another speaker. Thus utterances of the same words by several different training speakers must be used in the creation of the templates if input from a range of subsequent speakers is to be recognised accurately.

There are various possible ways to make use of multiple tokens (replications) of a word in making templates for it. One possible approach [48] is to use each token separately as a template. This, however, may result in an excessive amount of computation in the recognition process (unless the number of tokens is small, in which case they may not adequately cover all the possible variations) – since each unknown word must be compared with all the templates for all the reference words. Another possibility is to average the representations for all the tokens to produce a single word template. (The "averaging" of word representations is not altogether straightforward: they must first be aligned so that corresponding frames match up – using linear alignment, DTW or some other procedure [50,86,88] – before taking the average of the representations in each frame.) However, if the variations among different speakers' pronunciations are large, this will not provide adequately for all the variants. Better speed and accuracy in recognition can be obtained by more sophisticated techniques for choosing or creating templates from the reference tokens. A clustering analysis [48,53,54,55,94,100] can be carried out, forming clusters of similar tokens, and then a template can be formed for each cluster, by averaging the tokens in the cluster or by some other procedure. This clustering of tokens is done separately for each word in the vocabulary. Typically the recognition performance improves as the number of cluster templates used per word of the vocabulary is increased up to about 10, but this improvement levels off as further templates are added beyond that number [54]. Templates obtained by clustering give better recognition accuracy than the same number of templates chosen at random from among the training tokens [54,55]. (Clustering experiments with a speaker-trained recogniser have also been reported [56]. The results were qualitatively similar to those for the speaker-independent system, though fewer templates were required for optimal performance. The difficulty in applying

speaker-specific clustering in practice is that several training repetitions of the vocabulary are required.) An alternative to clustering, which has been found to yield better recognition results [102], is a procedure of selecting those training tokens required to obtain correct recognition of the training data set (after elimination of outliers): this results in condensed nearest neighbour classification (when used with a nearest neighbour decision rule, as described in section 2.3.8 below).

Where the vocabulary contains words which differ only in small portions, recognition errors are liable to occur because of differences between those parts of the templates which represent similar parts of words. For instance, if the vocabulary contains the words "stalactite" and "stalagmite", and the templates for these words differ in the initial "stala-" section, the input word "stalactite" may well be recognised as "stalagmite", because the linguistically insignificant difference between the templates in the (longer) "stala-" section outweighs the significant difference between "gm" and "ct". Such errors may be prevented [84,85] by combining the similar parts of templates for different words, so that, for example, the frame representations for the initial and final parts of the "stalagmite" template are identical to those for the corresponding parts of the "stalactite" template, and separate data for the two words are stored only for the distinguishing portions.

Experiments show [30,77,246] that, for recognition in noisy conditions, it is best to conduct the training in the same level of noise in which the unknown input words are to be spoken. If the characteristics of continuous background noise during the recognition session are analysed, this noise can be added to the original templates (obtained in quiet conditions) to improve the performance in the noisy environment [219,247].

2.3.3: Local path constraints

As mentioned in section 2.2, constraints of continuity and monotonicity are required for the time registration path, and it may also be desirable to impose restrictions on the steepness of the slope and the sharpness of changes in slope to prevent excessive distortion of the patterns being aligned.

These constraints can conveniently be combined in a specification of which points are allowed as predecessors to a given point. Let the possible sequences of recent preceding points at the point (m,n) be $P(1), \cdots, P(i), \cdots, P(r)$. (These sequences are "productions" [70]: the complete time registration path is constrained to be a concatenation of sequences each of which takes one of these r forms.) A condition $C(i)$ may be imposed on the permissibility of $P(i)$. For each permitted sequence $P(i)$, an accumulated distance $D^i(m,n)$ is calculated. Let the sequence $P(i)$ be defined by

$$P(i) = [(m_0^i,n_0^i), \cdots, (m_{K(i)-1}^i, n_{K(i)-1}^i)] \tag{2.9}$$

(where $K(i)$ is the number of points in the sequence $P(i)$). Then

$$D^i(m,n) = D(m_0^i,n_0^i) + \sum_{k=1}^{K(i)-1} w_k^i d(m_k^i,n_k^i) + w_{K(i)}^i d(m,n), \tag{2.10}$$

where $w_1^i, \cdots, w_{K(i)}^i$ are weighting factors (considered below in section 2.3.4). The accumulated distance $D(m,n)$ is then defined by

$$D(m,n) = \min\{D^1(m,n), \cdots, D^i(m,n), \cdots, D^r(m,n)\}. \tag{2.11}$$

For example, the Itakura constraints [18,51,60], as illustrated in figure 2.4(a), have $P(1) = [(m-1,n-2)]$; $P(2) = [(m-1,n-1)]$; and $P(3) = [(m-1,n)]$,

permitted on the condition $C(3)$ that the best path to $(m-1,n)$ (already found by the DTW algorithm) does not go through $(m-2,n)$. Note that in this case each of the sequences $P(1)$, $P(2)$ and $P(3)$ consists of a single point. These constraints force the average slope of the path to be no less than 0.5 and no greater than 2.0.

Other sets of constraints, denoted by Type I, Type II and Type III [60], are also shown in figure 2.4, with the sequences of preceding points permitted. In each case, no conditions $C(i)$ are imposed. (The naming of these sets of constraints is not consistent in the literature: in [142], the phrases "Type I" and "Type II" are interchanged.)

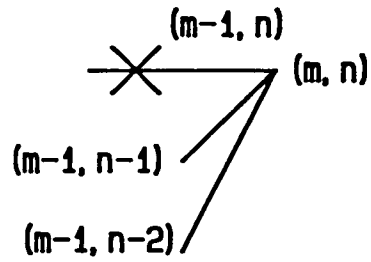
When conditions $C(i)$, such as $C(3)$ in the Itakura constraints, are allowed, the optimality principle of dynamic programming is no longer strictly valid. The Itakura constraints give only an approximation to the optimisation of the path which is afforded by the Type III constraints, since the optimal path may be excluded from consideration if it contains a step (1,0) starting at a point which can be reached by a partial path (which is not a part of the optimal complete path) ending with a step (1,0).

Comparative recognition tests [60,63] have shown that the Itakura constraints give a better rate of correct recognition than the Type III constraints. Very similar results to those using Itakura constraints were obtained [60] when Types I and II constraints were used.

All these constraints impose the same limits on the average slope of the path. Further examples of local path constraints, permitting different ranges of slopes, are given in [49]; the best results were obtained when the range of slopes permitted was from 0.5 to 2.0, as in the above instances. (When the search area in the input-reference plane is reduced to a narrow band about the linear path

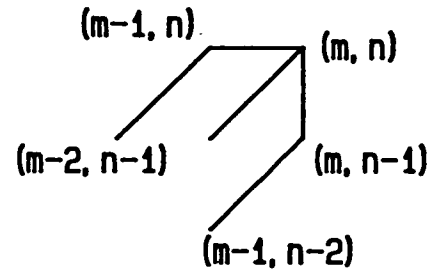
Figure 2.4: some local path constraints

(a) Itakura



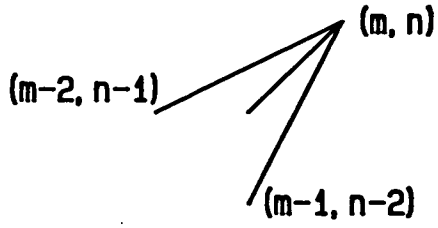
P(1) = [(m-1, n-2)]
P(2) = [(m-1, n-1)]
P(3) = [(m-1, n)]
C(3) = {path to (m-1, n) does not go through (m-2, n)}

(b) Type I



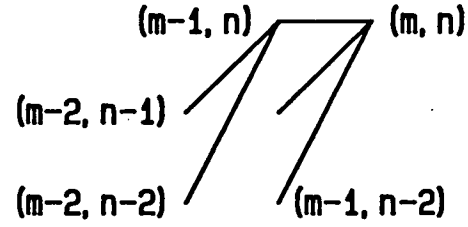
P(1) = [(m-1, n-2), (m, n-1)]
P(2) = [(m-1, n-1)]
P(3) = [(m-2, n-1), (m-1, n)]

(c) Type II



P(1) = [(m-1, n-2)]
P(2) = [(m-1, n-1)]
P(3) = [(m-2, n-1)]

(d) Type III



P(1) = [(m-1, n-2)]
P(2) = [(m-1, n-1)]
P(3) = [(m-2, n-2), (m-1, n)]
P(4) = [(m-2, n-1), (m-1, n)]

(see section 2.3.6), however, a restriction on the range of slopes may not be necessary [68].)

The Itakura and Type III constraints are not symmetric with respect to the reference and input directions. They could be modified by interchanging occurrences of "m" and "n" in the specifications of the sequences $P(i)$. This would correspond to having the reference frames in the horizontal direction and the input frames in the vertical direction in figures 2.2 and 2.4 [60]. Another modification [58] is to have the reference or input pattern in the horizontal direction according to which pattern has more frames. Having the longer pattern in the horizontal direction has the advantage that the steps which involve skipping a frame in the vertical direction are likely to occur less often.

2.3.4: Weighting of frame distances

The weight given to the frame distance at each point on a time registration path in calculating the accumulated distance, and hence the overall distance for the path, can be made to vary according to the slopes of the segments of the path near that point [49,60,65], and also according to other variables, such as the positions of the frames in the reference and input patterns [83,130,135,153].

Considering first the variation of weighting with path slope, this can be incorporated in the calculation of accumulated distances by means of the weighting factors w_k^i (where, for each value of i, k ranges from 1 to $K(i)$) mentioned above. Various schemes of weighting coefficients have been devised [60]. Two of the most commonly used are those called (c) and (d) in [60]:-

Scheme (c): weighting proportional to the distance moved in one chosen direction:

$$w_k^i = m_k^i - m_{k-1}^i \tag{2.12}$$

(if the input direction is chosen), where, for convenience of notation, $m_{K(i)}^i$ is defined to be m, and $n_{K(i)}^i$ is defined to be n.

Scheme (d): weighting proportional to the sum of the distances moved in the two directions:

$$w_k^i = (m_k^i - m_{k-1}^i) + (n_k^i - n_{k-1}^i). \tag{2.13}$$

As it stands, scheme (c) has the disadvantage that if a vertical segment occurs in a path a weighting of 0 will be assigned to the frame distance following that segment. This can be prevented [49,60] by smoothing (averaging) the weighting coefficients over the points $(m_1^i,n_1^i), \cdots, (m,n)$ for each production $P(i)$. In scheme (c) the sum of the weights for the whole path is M (the number of input frames) or, choosing the reference direction, N (the number of reference frames), and in scheme (d) the sum is $M+N$.

One modification [65] which can be applied to any weighting scheme involves using the weighting coefficient defined by a step in the path to multiply the distances at both the initial point of that step and its final point (rather than just the distance at the final point). This "trapezoidal" weighting has a smoothing effect, since the total weight on a frame distance is the average of the weights derived from the steps in the path preceding and following the point.

There are many other possible weighting schemes, such as those designed to penalise very steep or shallow slopes [23], but scheme (c) seems to have been the most frequently used. The choice of direction on which to base the weighting can be quite significant when this scheme is used [58,60]: it is better to base the weighting on the progress of the path through the input pattern rather than the

reference pattern [60], or the longer of the two patterns rather than the shorter one [58]. Such weights based on the input utterance have been found to give a better recognition rate than other options [60]. Weighting in the input direction also has the advantage that no template-specific normalisation of the word distances is required in making a recognition decision, because the total weight depends only on the length of the input word.

It is less usual to make the weighting depend on variables other than the slope of the path. However, it has been suggested [130] that the first and last parts of a word, for instance, may exhibit greater variability than the intermediate section, and in that case, to compensate for this, a smaller weight can be assigned to each frame distance in the corresponding parts of the time registration path than to each one in the rest of the path. (A more sophisticated adjustment of the weighting, which is made to depend on the rate of spectral change in the part of the word being matched, is described in [25]; but it is probably better to treat this as a modification of the frame distance function. The same applies to the weighted spectral slope metrics described in [27].)

A two-pass recognition procedure [61] and a subsequent modification of it [75] use special weights in a second-stage distance calculation, following the ordinary DTW matching, to improve discrimination between similar words. For each pair of words to be distinguished, a weighting function is used which takes larger values in those regions where the words differ most. These procedures require preliminary computation to determine the weighting functions, but have been found to improve recognition rates. They are rather similar in concept to the frame-specific and discriminatory distance functions [44,45] mentioned in section 2.3.1 above.

To penalise paths containing steps with slopes other than 1.0, additive penalties on such steps may be imposed, instead of multiplicative weights. In this case, the contribution made to the overall distance by the slope of each part of the path does not depend on the frame distances in that part of the alignment.

A method for deriving such penalties on the possible path steps for each part of a word individually, according to the frequency with which these steps tend to occur, has been proposed [70,83], and has been observed to improve discrimination between words differing mainly in their timescales, although having no beneficial effect on recognition in other cases. These reference-frame-dependent penalties are similar to the state-dependent transition probabilities in the Markov modelling approach described in section 2.5 below.

2.3.5: Endpoint constraints

Correctly locating the beginning and end of an utterance is not in general an easy task [15]. Speech sounds must be distinguished from background noise, and in particular from breath noise, clicks etc. made by the speaker at the end of the utterance.

The endpoint detection rule is commonly based on signal amplitude thresholds [62,87]. In this case, some sounds, notably voiceless fricatives such as the "f" in "five", may be classified as background noise, and thus excluded from the utterance. The risk of this may be reduced by adjusting the thresholds for distinction between speech and non-speech signals, but then there will be an increased tendency for background noise to be included as speech. It is possible to refine the endpoint detection procedure by incorporating a measure sensitive to fricatives, such as the zero-crossing rate [46]; but then any background hiss or other high-frequency noise occurring immediately before or after a word is liable

to be treated as part of the word. In general, the effectiveness of any particular set of parameters for endpoint detection will depend on the bandwidth of the signal, and on the spectral and temporal characteristics of the noise occurring. (For instance, the zero-crossing rate is not appropriate for use with telephone speech [62].)

To allow for errors in endpoint identification, the endpoint constraints on the time registration path may be relaxed to allow it to start and finish anywhere in specified regions at the beginnings and ends of the input and reference patterns, instead of only at the points (1,1) and (M,N) respectively [51]. This can be expected to result in some increase in the amount of computation required, as the region of the input-reference plane where frame distances are calculated will typically be increased.

Having a choice of pairs of endpoints complicates the identification of the optimal path. Firstly, the decision must be made whether to perform a separate time warp from each possible initial point or to include them all as points from which paths may come in a single warp (in which case the procedure will tend to favour paths from later initial points, since these paths will tend to have smaller accumulated distances). Secondly, whichever way that decision is made, at the end of the warping process there will be a number of complete paths to choose from, with different final points, and possibly also different initial points. Simply choosing the path with the smallest overall distance has the disadvantage that the paths are of different lengths and there will be a bias towards the shorter paths, and so it may be desirable to normalise the overall distances for path length before comparing them [51]. (The edge-free staggered array DP algorithm [38,73] avoids these problems, where the weighting is proportional to the sum of the distances moved in the input and reference directions, by allowing paths to start and finish anywhere on selected diagonals of slope -1 (where

$m+n = 2$ and where $m+n = M+N$, respectively). This requires the inclusion of sections of the reference and input signals beyond the detected word endpoints.)

An endpoint relaxation technique can improve the performance of a recogniser [51,63], but, depending on the characteristics of the vocabulary used, and of the acoustic background conditions and endpoint detection procedure, it may also introduce errors which outweigh the improvements [91].

Another endpoint modification technique [91,93,160] is a procedure in which a silence or noise frame, which can be matched repeatedly with successive frames of the input speech, is appended to each end of each template, and extra frames of the input signal beyond the detected utterance endpoints are used. This reduces the incidence of errors due to exclusion of parts of the input utterance, while allowing any non-speech frames in the extra input regions to be matched to the initial and final silence frames and so generate no errors (since the distance added by the matching of a given input frame to silence will be the same for each template, as the silence frames of all the templates are the same). Like the edge-free version of staggered array DP [38,73], this technique eliminates the problem of unequal total weights on partial paths from different starting points – since a silence-frame distance is added to the accumulated distance for every input frame preceding the effective start of the matching, and so the total number of local (silence-frame and template-frame) distances is the same on every path. A modification has been formulated [95] in which no prior detection of input word endpoints is required.

Another possibility, to improve endpoint alignment without calculating so many extra frame distances, is to do some preliminary testing at the beginning and end regions and, in each of these regions, to choose the alignment so that the local match is optimised. This procedure [58] gives new initial and final

points for the time registration path, which is then determined in the usual way with these new points replacing (1,1) and (M,N). Thus the extra distance calculations involved are only in small initial and final regions. In a comparative test [91] this technique was found to increase recognition accuracy slightly, but not as much as the technique with silence frames described above.

2.3.6: Global (search area) constraints

When the endpoints of the time registration path are specified exactly, the area of the input-reference plane in which the path may lie is a parallelogram determined by the maximum and minimum slopes allowed by the local path constraints. When the endpoints are variable, it is a significantly larger polygon. It is possible to impose further restrictions on the area to be searched for possible paths; this reduces the number of local and accumulated distance computations, and may also improve the accuracy of the recogniser by preventing excessive distortion.

One simple form of area restriction is to exclude points more than a fixed number of reference (or input) frames away from the straight line joining (1,1) to (M,N) [58,68]. An even simpler method is to use the line of slope 1.0 from (1,1) instead of the line from (1,1) to (M,N), so that (m,n) is excluded if $|m-n|$ exceeds some constant value ε [49,60,68]. This has the disadvantage that if the durations of the reference and input words differ by more than the chosen value ε there will be no permissible time registration path.

A more sophisticated area restriction method is the adaptive one used in the UELM (unconstrained endpoints, local minimum) algorithm [51]. Here the values of n considered for a given m are those not more than ε away from $\bar{n}$, where the value of $\bar{n}$ is chosen to minimise $D(m-1,\bar{n})$. In this case no con-

straint can be imposed on the final point: if the matching is successful, the algorithm will find the final point itself.

2.3.7: Word length normalisation techniques

There are certain advantages in having the same number of frames in each of the two words being matched together. A simple search area restriction (section 2.3.6) can sensibly be applied, as can weighting coefficients which penalise steps of slopes other than 1.0 (since the optimal path can be expected to be fairly close to the linear path from (1,1) to (M,N), which in this case ($M=N$) has slope 1.0).

2.3.7.1: Linear word length normalisation

If the durations of the words are determined before the frame representations are computed, it is possible to adjust the frame separations so as to have the same number of frames in every word [47] (whether a reference template or an input word). This, however, introduces a time-lag in the operation of the recogniser, and requires the calculation of frame representations to be adaptable for varying frame separations, although with overlapping windows the latter is not such a significant point since the separation can be changed while keeping the window length constant without leaving gaps of unused data between windows.

Another way to adjust the numbers of frames is a linear interpolation technique [60]. In this method, frame representations are computed with a standard time separation in the usual way, and then once they have all been stored (and the duration of the word is known) they are replaced by a preset number of frame representations at equal intervals throughout the word. These frame

representations are calculated from the original representations by linear interpolation. The reference templates are stored in length-normalised form, so that only the input word needs to be normalised before each word recognition. The time warping cannot start until the whole input word has been read in, but the time-lag will be less than with the preceding method provided that the interpolation procedure takes less time than the calculation of frame representations from raw sampled input. Also, for this interpolation procedure only the frame representations, rather than the individual sample values, must be stored until the word has been fully read in, so that it is likely to be more economical in use of memory than the preceding method. Experiments show [60] that, when word lengths are normalised, imposing a restriction $|m-n| \leq \varepsilon$ can improve the recognition slightly, rather than degrading it.

Another way to apply the interpolation procedure would be to leave the input frame representations as they are and apply the interpolation to the reference words to normalise each of them to the length of the input word. This would have the advantage of adapting the number of frames used so that more were used for a long input word than for a short one, thus possibly achieving a better combination of recognition performance and economy in frame distance calculations, but would require the interpolation to be done each time for all the reference words, rather than just for the one input word.

2.3.7.2: Trace segmentation

A trace segmentation procedure [76,80,81] has some similarities to the word length normalisation technique outlined above. It uses linear interpolation to form new frame representations from the original ones, and it results in word representations of a fixed length. However, there is an important difference, in

that the new representations are not regularly spaced in time, but are chosen so that the amount of spectral change in the speech signal from one representation to the next is constant over all parts of the word.

Bandpass filter energies are extracted at regular time intervals in the usual way, and extra frames representing silence are appended to the utterance, one at the beginning and one at the end. If the vectors of filter outputs for the successive frames, including the initial and final silence frames, are denoted by $\mathbf{x}(0),\mathbf{x}(1),\cdots,\mathbf{x}(J)$ (where the number of non-silence frames is $J-1$), then

$$T = \sum_{j=1}^{J} d(\mathbf{x}(j-1),\mathbf{x}(j)), \tag{2.14}$$

where T is the total length of the trace in Q-dimensional space (Q = number of filters) formed by joining the points defined by consecutive frame vectors. (Here d is a distance function for vectors, such as the absolute value distance.) The trace is divided into S segments each of length $\frac{T}{S}$, where typically S is about $\frac{J}{3}$. The vectors defining the segment boundaries are computed, and these are used as the new frame vectors (together with the initial and final silence vectors).

This trace segmentation procedure followed by the usual DTW alignment of the (new) vector sequences makes up the word matching method called DYPATS (dynamic programming after trace segmentation) [76]. It not only adjusts each word to a standard number of frames but also gives greater weight to parts of a word where spectral change is occurring, which is probably more sensible than giving equal weight to equal segments of time [25]. Moreover, it can result in considerable savings in computation and data storage, especially when used in conjunction with a strict search area restriction, while maintaining or improving

on the level of recognition accuracy obtained with straightforward DTW [76].

Although the formulation above is for a recognition system using bandpass filter representations, without logarithmic transformation [76], trace segmentation can be used with other acoustic representations [80,81,152]. The appending of silence frames may not be appropriate in this case.

Another form of trace segmentation [80,81] divides the trace into segments of a predetermined constant length, rather than into a predetermined number of segments. This has the advantage that processing can start as soon as the first input frame vectors are calculated, but does not result in normalisation of all words to the same number of frames.

There are various other acoustically-based non-linear segmentation techniques [66,69,80,82,168]. These rely, like trace segmentation, on measuring distances between nearby frame vectors of a word (but not always successive ones) and applying a threshold to determine where to start a new segment.

Other possible methods of deriving a sequence of vectors to represent a segmented word, instead of interpolation at the segment boundaries, are selection of the nearer of the two neighbouring input vectors at each segment boundary [81] and averaging of all the vectors in each segment [152]. (An experimental comparison of these methods will be given below, in chapter 4.)

2.3.8: The recognition decision

Once an unknown word has been read in and compared by the DTW procedure with all the reference templates, there will be a list of overall distances (perhaps normalised for template length, depending on the weighting scheme adopted) from the input word to the various templates. (It has been found beneficial [98] to normalise the distance for each word according to the word's

variability, as determined from a statistical analysis; this requires a number of reference utterances for each word of the vocabulary.)

The way in which these distances are used will depend on various aspects of the system. If there are several templates for each vocabulary word, it may be decided to assess a reference word by the distances to all or several of its templates rather than just by the distance to the nearest of them. Decision rules for this case are discussed below, in section 2.3.8.2. But even where there is just one template for each word there are different ways to use the distances.

2.3.8.1: Decision procedures in a single-template system

In a recogniser with one template per word, the most straightforward decision procedure is to recognise the input utterance as the word whose template gives the smallest distance.

A modification is to make a recognition decision like this only if the distance is less than some fixed value [18], or only if it is less than the second-smallest distance by at least some fixed difference or ratio [54], or only on both of these conditions [126], and to give a "reject" or "no recognition" response if the condition is not met: this reduces the rate of wrong recognitions, at the expense of a rejection rate which will generally include some rejections of correct nearest templates. This modification is appropriate for a system in which high reliability of recognised words is desired and the operator does not mind having to repeat some words when they are not clearly recognised the first time.

Another option is for the output from the template matching process to consist not just of a single best-matching word but of an ordered list of the best several candidates [54,55]. This is particularly useful in a system where there are restrictions on what word sequences may occur (due to the syntax or format

of the input) [126,127], since these restrictions can be used to eliminate some of the words listed as possible by the template matching process at each position in the sequence. This procedure can be applied, for example, in a directory assistance system, to the recognition of names spelt out (so that the words to be recognised are letter names), with the restriction on word sequences being that the name formed must be in the directory [12,128].

2.3.8.2: Multiple-template decision rules

In the case of several templates for each word, there are various decision rules that can be applied to define the best identification for an input word, among them the nearest neighbour (NN), K-nearest neighbour (KNN) and majority vote decision rules.

The NN rule is the simplest: the input word is recognised as the word corresponding to the template whose distance from the input pattern is smallest. This is similar in implementation to the procedure described above for the case where each word in the vocabulary is represented by just one template.

The KNN rule [54,55,90] takes the average, for each word in the vocabulary, of the distances from the input word to the K nearest templates of the vocabulary word, and then chooses the vocabulary word for which this average is smallest. (With 10-12 speaker-independent templates per word, 2 or 3 is a suitable value for K [12,54,55,90].) Notice that the NN rule is a particular instance of the KNN rule, namely that in which $K = 1$.

The majority vote rule (sometimes confusingly called KNN [47,89]) takes into account the K templates (of whatever words) nearest to the input pattern, and chooses the word represented by the greatest number of these K templates. In the event of a tie among different words, the NN criterion is used to decide

among them. (K will usually be larger for this rule than for the KNN rule: the value $K = 7$ has been found to give good results [47,89] when there are 12 or 14 templates per word.) This too reduces to the NN rule when $K = 1$, and indeed also when $K = 2$.

Many other decision rules could be devised, such as modifications of the KNN rule using weighted averages, or combinations of majority vote rules with several different values of K. A procedure using NN or majority vote, depending on features of the distribution of templates around the input pattern, was found [47] to give better recognition performance than either of the two rules used alone.

2.4: Modifications to reduce computation and storage requirements

DTW algorithms are computationally expensive, especially when the vocabulary is large and each input word has to be compared with every one of the templates. Various means of reducing the amount of computation, and in some cases also the storage requirements (for reference data or for quantities used in the algorithm such as accumulated distances), have been proposed.

2.4.1: Template elimination procedures

One way to eliminate a good deal of the computation is to impose some sort of accumulated distance threshold on each template as it is being matched with the input word, and to abandon the matching process if the threshold is exceeded [12,18,54]. The accumulated distance threshold typically takes the form $Am + B$, where m is the current input frame number [12,54].

Another procedure [47] which has a similar effect is to carry out the matching with the input word for all the templates in parallel and, at certain stages, to exclude from consideration prespecified proportions of the templates (choosing those with the largest accumulated distances).

A beam search strategy [64,66] can be adopted, in which all the templates are matched in parallel, and at each input frame only those templates – or only those partial paths, in whatever template – which have accumulated distances less than a fixed threshold above the best current accumulated distance are retained. This has an advantage of adaptiveness over the fixed absolute threshold method, and, unlike the fixed-proportion exclusion method, it allows the number of templates under consideration at each stage to depend on whether there are many templates with accumulated distances close to the current minimum.

The amount of computation required to recognise each word can be reduced by such methods by a factor of 3-20, depending on details of the implementation and on the vocabulary [18,47,64,66].

A branch-and-bound or best-first strategy [64] involves continuing, at each stage, the path (in whatever template) which has the least accumulated distance. Thus not all paths under consideration at any stage will necessarily have reached the same input frame. A pruning technique can then be applied, by which any path which falls behind the longest current partial path (in the input direction) by a set number of frames is abandoned. This method of pruning has the same advantages mentioned above for beam searching. A drawback of the branch-and-bound approach is the large number of accumulated distance comparisons required to identify the paths to be extended.

If some words in the vocabulary are very dissimilar to the input word, it may not be necessary even to begin the DTW matching for these words: they can be eliminated by some simpler procedure before they get to the DTW stage. For instance, a "match limiter" has been employed [229] which makes an initial comparison of each template with the input, using only duration and three averaged spectra to represent each word; only the K words (for some fixed value of K) with the best scores, out of a much larger vocabulary, are passed on to the second stage which performs standard DTW matching. Procedures similar in principle to this have been devised using a variety of other comparison methods to obtain the preliminary scores [230,231,232,233,234,235,236]. A threshold on the distance in the preliminary match, possibly depending on the distance obtained for the best-scoring template, can be applied to decide which words are passed on, as an alternative to specifying a fixed number of words [231,232]. (Options in a two-stage comparison procedure of this sort are discussed in more detail in chapter 3 below. The implementation of such a procedure, which can be extended to more than two stages, is described in chapter 4.)

In a multiple-template system, another option is to match initially only a certain number of the templates for each word, and on the basis of the distances obtained to select a few words for matching of all their templates [99].

2.4.2: Reductions in computation and storage per template

Another approach, which can be combined with the techniques already described, concentrates not on eliminating templates but on reducing the amount of computation to be done for each template in aligning it with the input word. Some examples of this sort of reduction have already been examined, namely the restrictions on the area in which paths are allowed. Another tech-

nique which reduces the amount of searching done to find the time registration path is an ordered graph search (OGS) [67,74] – which is rather similar to the branch-and-bound technique described above, but applied to each template separately. This procedure reduces the number of frame distances calculated, but requires considerably more computation for other parts of its execution than the standard DTW search: thus its usefulness depends on details of the implementation.

2.4.2.1: Reduction of the number of frame representations

Both the computation of frame distances for each word pair matched and the storage requirement for templates can be reduced by reducing the number of frame representations stored for each word. One way to do this [52] is (in contrast to storing separately the representations for successive frames where these are similar) to store only the first frame representation for a steady section of speech. Representations for succeeding frames are treated as being identical to the first one as long as they differ from it by less than some preset amount. In the system in which this procedure was implemented, it was found that the reference storage requirements could be reduced by a factor of 2, and the frame distance calculations reduced by a factor of 4 (since the reduction was applied to both input and reference frames), without significant loss of recognition performance [52].

Various methods of variable frame rate coding and acoustic segmentation have been devised [57,66,69,76,80,81,82,168], to reduce the number of (frame or segment) representations per word. Information about the durations of the segments represented may be stored, and used to control the matching of segments [57,66,168] or to determine how many times each representation should be

repeated in a "segment expansion" procedure [69]; the latter case is very similar to the procedure of [52] described above. If there is no expansion by repetition of representations, there is a saving in accumulated distance computation, as well as a reduction in the number of frame distances to be calculated, but in this case the accuracy of the recogniser may deteriorate as the number of segments per word is reduced [69].

The DYPATS process [76] described in section 2.3 reduces computation and storage – because $S < J$ and so the number of frames in each word is reduced; and also because the trace segmentation stage provides part of the time warping required (in the case of reference and input being the same word) and so the DTW search can be restricted to a narrow band around the diagonal from (0,0) to (S,S) without much loss of recognition accuracy.

2.4.2.2: Vector quantisation

The technique of vector quantisation (VQ) [32,78,79,80,104,164] can be used to reduce the reference storage requirements and the number of frame distances to be calculated, particularly if the number of templates is large. This technique generally results in some degradation of the recognition performance.

Vector quantisation can be one-sided or two-sided [80,164]. In the case of one-sided quantisation, a codebook is constructed, consisting of vectors of speech frame parameters, and each frame of each template is represented by one of the codebook vectors, usually the one nearest to the actual frame vector. When an input frame vector is received, it is matched with all the codebook vectors, and the distances thus obtained are used in place of the true input-reference frame distances in the DTW algorithm. The templates are stored as sequences of codebook vector indices, rather than sequences of actual vectors; the codebook vec-

tors themselves are stored separately. In two-sided VQ, both reference and input frames are represented by vectors from the codebook. This has the advantage that the distances for all pairs of vectors in the codebook can be stored as a table, so that they do not have to be calculated afresh when they are required; some computation will have to be done, however, to identify which codebook vector should be used to represent each new input frame. (This need not involve matching the input vector with all the codebook vectors: a faster VQ technique such as binary tree coding [164,171] can be used – though this may reduce the recognition accuracy.) Depending on the size of the codebook, quite a large amount of memory may be required for the table of distances.

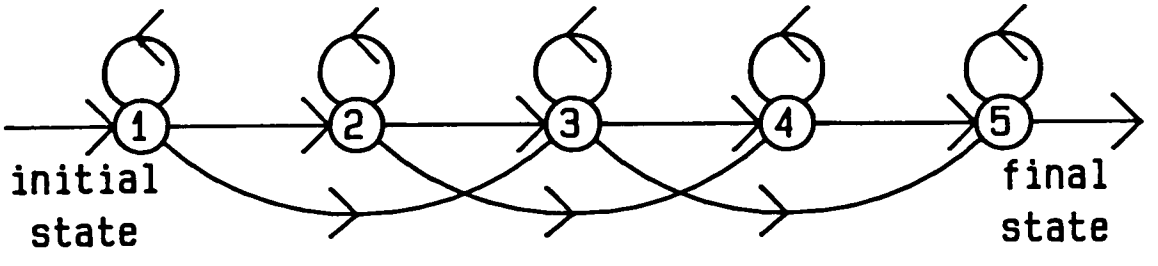
2.5: Dynamic programming applied to hidden Markov models

A more sophisticated approach to speech recognition is to construct a statistical model (rather than a template or templates) of each word in the vocabulary, and recognise each input word as that word of the vocabulary whose model assigns the greatest likelihood (probability or probability density) to the occurrence of the observed input pattern. One type of statistical model [7,103,104,105,107,154,171] is the hidden Markov model (HMM) described below. If such a model is adopted, a form of dynamic programming algorithm, called the Viterbi algorithm [5], can be applied to calculate, for each word's model, the likelihood for the optimal matching of the input to a sequence of states of the model [5,7,103,104,107,171].

The essential idea of the HMM approach is that each word in the vocabulary is represented by a set of states, including at least one initial state and at least one final state, with probabilities of transitions from state to state, and for each state a probability distribution for the emission of a vector of acoustic

parameters. (The states and transitions can be shown in the form of a network, as in figure 2.5.) When the word is spoken, the process is assumed to be in an initial state when the word begins, and then to make state transitions at time intervals equal to the separations between speech frames, in such a way that it is in a final state when the word ends. At each time frame, a vector is emitted whose probability distribution is that associated with the current state. (An alternative formulation [7,105] has the emission probabilities associated with the transitions rather than with the states.) The states themselves are not assumed to be observable, but only the emitted frame vectors which are probabilistically related to the states – hence the use of the word "hidden". (The name "Markov" is applied because the state transitions form a first-order Markov chain: that is, the probabilities of transitions from a given state depend only on the identity of that state, and not on which states the process has been in at previous times.)

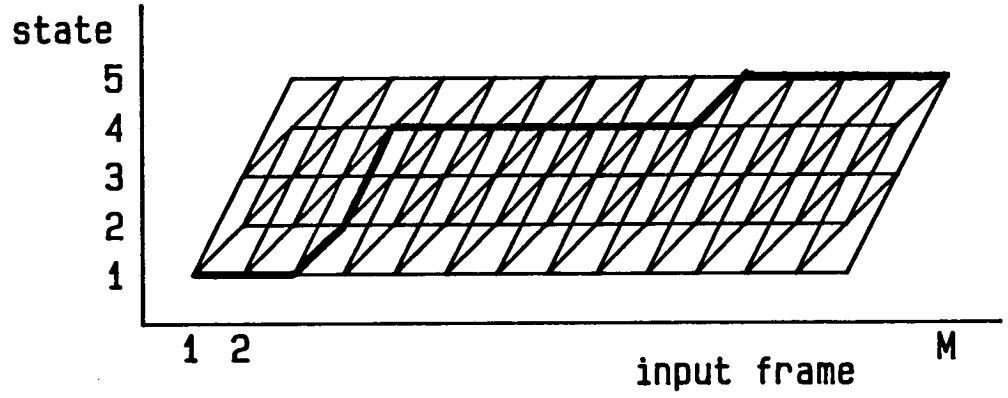
Figure 2.5: state transition network for a Markov model



To apply the Viterbi dynamic programming algorithm to the matching of a sequence of input frames to a given HMM, the states of the Markov model are arranged along the reference axis, as in figure 2.6, where the possible transitions at each input frame are shown in the form of a trellis [5] and the optimal alignment is marked by the heavy line. In the following discussion of the algorithm, it will be assumed that the model to be matched has N states, including one initial state, state 1, and one final state, state N.

At each step, the path must advance exactly one frame in the input direction, and can move in the reference direction to any state to which there is a possible transition from the preceding state. Thus, for instance, if the allowable transitions to state n are from states $n-2$, $n-1$ and n, the possible predecessors of the point (m,n) are $(m-1,n-2)$, $(m-1,n-1)$ and $(m-1,n)$. Note that the

Figure 2.6: alignment of input to states of a Markov model using the Viterbi algorithm



Optimal state sequence: 1, 1, 1, 2, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5.

possible steps in the path are not necessarily the same at all points (even ignoring endpoint effects): they depend on which transitions exist among the particular states in the part of the Markov model that the path has reached. (In the case illustrated, however, the model has a regular structure, and so the possible path steps exhibit a corresponding regularity.)

Assuming that all transition and emission probabilities are independent (which is not a very realistic model of the structure of speech, but is convenient for mathematically tractable modelling), the probability to be calculated is the product of the probabilities (and probability densities, if the emission probability distributions are continuous) of the individual transitions and emissions occurring when the observed input word is generated by the model. (In the paragraphs below, "emission probability" will be used loosely to mean "emission probability or probability density", to allow for the possibilities of discrete or continuous distributions.) Let the transition probabilities be defined by

$$a_{ij} = P(\text{state } j \text{ at time } t \mid \text{state } i \text{ at time } t-1) \tag{2.15}$$

(where the times are measured in frame intervals), and let b_i be the vector emission probability distribution associated with state i, so that

$$b_i(\mathbf{x}) = P(\text{vector } \mathbf{x} \text{ emitted} \mid \text{state} = i) \tag{2.16}$$

– where i and j range from 1 to N. Then the probability, given this model, that the state sequence $(i_1, \cdots, i_M)$ occurs and the vector sequence $(\mathbf{x}_1, \cdots, \mathbf{x}_M)$ is emitted is

$$P(i_1, \cdots, i_M; \mathbf{x}_1, \cdots, \mathbf{x}_M \mid \text{model}) = b_{i_1}(\mathbf{x}_1) \prod_{m=2}^{M} (a_{i_{m-1}i_m} b_{i_m}(\mathbf{x}_m)). \tag{2.17}$$

The task of the Viterbi algorithm is to find the state sequence $(i_1, \cdots, i_M)$ which maximises this probability for the observed (input) acoustic vector sequence $(\mathbf{x}_1, \cdots, \mathbf{x}_M)$, subject to the constraints that $i_1 = 1$ and $i_M = N$. This state sequence corresponds to the optimal path through the trellis (as in figure 2.6).

For computational reasons, it is convenient to use the logarithms of the probabilities (and probability densities)in the dynamic programming procedure, rather than working with the probabilities themselves. Products of probabilities will then be transformed to sums of log probabilities. Furthermore, to demonstrate the underlying similarity of the Viterbi and DTW algorithms, it is best to use the negatives of the log probabilities. Thus the maximising of the product of probabilities in (2.17) is replaced by the minimising of the corresponding sum of negative log probabilities. Define the following negative log probabilities:-

$$u_i(\mathbf{x}) = -\log b_i(\mathbf{x}); \tag{2.18}$$

$$z_{ij} = -\log a_{ij}. \tag{2.19}$$

Then the negative log probability of the first m states and emitted vectors is given by

$$U(m,n) = u_{i_1}(\mathbf{x}_1) + \sum_{k=2}^{m} (z_{i_{k-1}i_k} + u_{i_k}(\mathbf{x}_k)), \tag{2.20}$$

(where $i_m = n$). From (2.17-2.20), with a logarithmic transformation of the product of probabilities in (2.17), it can be seen that

$$U(M,N) = -\log P(i_1, \cdots, i_M; \mathbf{x}_1, \cdots, \mathbf{x}_M \mid \text{model}). \tag{2.21}$$

The correspondence [13,14] between the Viterbi algorithm and DTW (expressed in the notation of sections 2.2 and 2.3 above) is then as follows.

DTW	*Viterbi*
frame distance: $d(m,n)$	negative log emission probability (density): $u_n(\mathbf{x}_m)$
penalty on step $(1,n-n')$ in path (see end of section 2.3.4)	negative log transition probability: $z_{n'n}$
accumulated distance: $D(m,n)$	negative log probability of sequence of transitions and emissions so far: $U(m,n)$
word distance: $D(M,N)$	negative log probability of completed sequence: $U(M,N)$

The dynamic programming procedure to find the optimal state sequence is initialised by setting

$$U(1,1) = u_1(\mathbf{x}_1) \tag{2.22}$$

(with $U(1,n)$ infinite for $n > 1$). Then, if the possible predecessors for state n are states $n_1, \cdots, n_i, \cdots, n_r$, the recursion is

$$U(m,n) = \min\{U(m-1,n_i) + z_{n_i n} \mid 1 \leq i \leq r\} + u_n(\mathbf{x}_m). \tag{2.23}$$

(Compare equations (2.1), (2.2), (2.8) and (2.9).)

The Viterbi algorithm reduces to a simple form of DTW (with $P(i) = [(m-1,n_i)]$ for each i, weighting scheme (c) and an appropriate distance function) if a state is defined for each template frame and all the transition probabilities are equal – apart from an additive quantity, depending only on the input frame number, due to the addition of negative log transition probabilities. The emission probability distribution for each state in the HMM corresponds to a frame vector in the template and the associated frame distance function.

In particular, if the emission probability distribution for state n is multivariate Gaussian with mean $\mathbf{y}_n$, equal variances σ^2 for all components and zero covariances, then u_n has the form

$$u_n(\mathbf{x}) = Q\log(\sqrt{2\pi}\,\sigma) + \frac{\sum_{i=1}^{Q}(x(i)-y_n(i))^2}{2\sigma^2}, \tag{2.24}$$

where Q is the dimension of the vectors and (for $i = 1, 2, \cdots, Q$) $x(i)$ and $y_n(i)$ are their ith components. Then, if all allowable transitions to state n have the same negative log probability z, equation (2.23) becomes

$$U(m,n) = \min\{U(m-1,n_i) \mid 1 \le i \le r\} + z + Q\log(\sqrt{2\pi}\,\sigma) + \frac{\sum_{i=1}^{Q}(x_m(i)-y_n(i))^2}{2\sigma^2} \tag{2.25}$$

(where $x_m(i)$ is the ith component of $\mathbf{x}_m$): this is the equation for the DTW recursion with the squared Euclidean metric, apart from the additive constant $z + Q\log(\sqrt{2\pi}\,\sigma)$ and the constant factor $\frac{1}{2\sigma^2}$. Indeed, if

$$\sigma = \frac{e^{-z/Q}}{\sqrt{2\pi}}, \tag{2.26}$$

the additive constant disappears, and, by scaling the input and reference vectors to eliminate the factor $\frac{1}{2\sigma^2}$, equation (2.25) can be made identical to the appropriate case of the general DTW equation (2.9). (To make this hold true for all states of the model, while retaining a fixed scaling of the vectors, the value of z must be held constant: thus, if the number of possible transitions from state n is not the same for all n (excluding N if transitions from state N are not allowed to occur), the stochastic constraint on the transition probabilities (that their sum must be 1.0) will not be perfectly satisfied.)

Similarly, HMM formulations of other forms of DTW algorithm can be devised. One corresponding to the Type III constraints [159], for instance, involves two copies of each state (strictly, two states with the same emission probability distribution but different transitions); the second copy is used when a state is repeated (corresponding to a step (1,0) in the path). Weighting schemes other than scheme (c) can be implemented by having copies of each state with differently scaled emission probability distributions (though this again involves a departure from the strictly stochastic framework of modelling). Frame distances other than the squared Euclidean metric can be generated by using other forms of emission probability distribution [13]. (The relation between HMMs with Gaussian autoregressive probability densities and DTW with an LPC-based distance measure is developed in detail in [14].)

Thus DTW is in principle a special case of the Viterbi algorithm. The peculiarities of DTW are that the number of states (template frames) per word tends to be larger than in other word recognition techniques using the Viterbi algorithm, and indefinite repetition is not usually permitted, and that the variability of the transition and emission probabilities from one state to another is more constrained.

The Viterbi algorithm does not, strictly speaking, obtain the likelihood (probability or probability density) of the observed input speech given the HMM for the hypothesised word: it obtains the likelihood of the more restricted event that, given the HMM, the observed input vectors are emitted from a specific sequence (the optimal sequence) of states. The overall likelihood of the observed input vectors, given the model, is the sum of the likelihoods with specified sequences of states. The above algorithm can be adapted to calculate this by first returning to the multiplicative domain (multiplying probabilities, instead of adding their negative logarithms), and then at each point adding the accumulated partial-path probabilities, instead of selecting the maximal one, before multiplying by the local likelihood. (This adapted algorithm is related to the forward-backward algorithm described below [103]: the quantity computed at each point is the forward probability.) This calculation of the overall likelihood (called Baum-Welch scoring [103]) is more expensive computationally than Viterbi scoring, because of the multiplications involved.

As already mentioned, the emission probability distribution for each state can be discrete or continuous. In the discrete case [103,104,105,171], the output of the model is assumed to be a sequence of symbols from a finite alphabet. To cope with input consisting of acoustic parameter vectors, vector quantisation (see section 2.4.2) must be applied; then the HMMs are matched with the sequence of VQ codebook indices derived from the input. In the continuous case, the distribution associated with each state is typically some form of Gaussian distribution, or a mixture of such distributions [107,110,112,113] – though a system incorporating non-parametric distributions, obtained using a Parzen estimator, has been reported [114]. Continuous distributions have been found to yield better recognition performance than discrete ones [107,110]. (The reliable estimation of continuous mixture distributions requires a large amount of training

data; but an improvement in performance over the discrete case can be attained, without such a large training requirement, by the semi-continuous HMM technique, in which one-sided VQ is applied, and the vector probability distributions corresponding to the codebook vectors are estimated when the codebook is formed and are used in the recognition phase to compute likelihoods for input vectors [120,121].)

For the case where the emission probability distributions are discrete and are associated with transitions rather than states, an alternative formulation [8] has separate transitions, with appropriately adjusted probabilities, corresponding to all the possible emitted symbols; in this case the Markov model is explicit, rather than hidden, but essentially the same parameter estimation and recognition algorithms can be applied.

Models for words which are similar in some (initial and final) parts can be combined into a single model, thus economising on storage and computation – rather as the templates for such words are combined in the discriminative technique described in section 2.3.2 above. The problem of computing appropriate transition probabilities for such models is addressed in [111].

Standardised forms of state transition network have been devised. One such is the Bakis model [156,171], which allows a state to be repeated indefinitely or to be omitted: thus the local path constraints are those of the simple example in section 2.2 above. (The transition network in figure 2.5 is for a five-state Bakis model.)

A comparison of digit recognition results using speaker-trained HMMs with different structures is given in [117]. The best results were obtained when the number of states was large (e.g. 20) and arbitrary left-to-right transitions were permitted. Most other results reported have been for models with fewer states –

typically about five [103,104,105,107,110,111,112,113].

Markov models have been used particularly in connected word recognition [8,159,196], as described in section 2.7 below.

2.5.1: Estimation of HMM parameters

The training procedure for each word of the vocabulary is rather more complex for an HMM-based recogniser than for a template-based one. First, the number of states to be used to represent the word must be decided, and allowable transitions between states must be defined [103,104,117]. Then the emission and transition probabilities must be estimated; this can be done in various ways.

One method of estimating the transition probabilities [10,156,159,171] is to apply the forward-backward algorithm (described below) or the Viterbi algorithm iteratively to the training data for the word being modelled. Before this is done, each state in the word model must be assigned a probability distribution for the emission of frame vectors. This is done by analysis of the distribution of frame vectors from the training data which might (as indicated, perhaps, by least-squares segmentation of training utterances of the word [131,133]) correspond to that state. Then the transition probabilities are estimated by starting with (for example) equal probabilities for all transitions from each state, applying the forward-backward or Viterbi algorithm to each training token of the word and adjusting the probabilities to accord with the frequencies of occurrence of the transitions. The emission probability distributions can be reestimated at the same time by accumulating statistics on the vectors in the training data that are associated by the algorithm with each state. This procedure can be iterated several times.

The forward-backward algorithm [8,10,14] is a recursion procedure which accumulates statistics from all possible alignments of the data with state sequences of the model, rather than only the alignment with the optimal state sequence which is found by the Viterbi algorithm. The iterative training algorithm which uses these statistics is called the Baum-Welch algorithm [103,104]. Statistics on the vectors occurring with each state are assigned weights in the training procedure which correspond to the probabilities with which the state-vector combinations occur. Similarly, the reestimated probabilities for transitions from a state are proportional to the weighted sums of occurrences of these transitions over all possible alignments – where the weights are again the probabilities assigned to the occurrences by the model with its existing parameters. To compute these weighted sums efficiently, they are formulated in terms of forward and backward probabilities, which can be calculated by the forward-backward algorithm.

The forward probability for state n and input (training) frame m is defined by

$$\alpha(m,n) = P\{\mathbf{x}_1, \cdots, \mathbf{x}_m; \text{ state} = n \text{ at frame } m \mid \text{model}\} \tag{2.27}$$

and the backward probability is defined by

$$\beta(m,n) = P\{\mathbf{x}_{m+1}, \cdots, \mathbf{x}_M \mid \text{model; state} = n \text{ at frame } m\}. \tag{2.28}$$

Note that the probability of the utterance, given the model, is equal to $\alpha(M,N)$. Thus the probability of state n at frame m, given the model and the training data, is

$$s(m,n) = \frac{\alpha(m,n)\beta(m,n)}{\alpha(M,N)}. \tag{2.29}$$

Likewise, the probability of a transition from state n to state n' at frame m is

$$t(m,n,n') = \frac{\alpha(m-1,n)a_{nn'}b_{n'}(\mathbf{x}_m)\beta(m,n')}{\alpha(M,N)}. \tag{2.30}$$

By adding the quantities computed by (2.29) and (2.30) over all values of m, overall probability-weighted frequencies of states and transitions can be obtained. These statistics can be accumulated over multiple training utterances, and used to reestimate the transition probabilities. Similarly, probability-weighted statistics for occurrences of acoustic vectors can be computed and used to reestimate the emission probability density parameters.

The computation of the forward probabilities starts with the values of $\alpha(1,n)$ for all state numbers n: $\alpha(1,1) = b_1(\mathbf{x}_1)$ and $\alpha(1,n) = 0$ where $n > 1$ (assuming that state 1 is the unique initial state). It then proceeds by the recursion

$$\alpha(m,n) = (\sum_{i=1}^{r} \alpha(m-1,n_i)a_{n_i n}).b_n(\mathbf{x}_m), \tag{2.31}$$

where, as in (2.23), the possible transitions to state n are from states $n_1, \cdots, n_r$. The computation of the backward probabilities starts at the end of the utterance, where $\beta(M,N) = 1$ and $\beta(M,n) = 0$ for all other values of n (assuming that state N is the unique final state); the recursion for backward probabilities is

$$\beta(m,n) = \sum_{i=1}^{k} a_{nn^i}b_{n^i}(\mathbf{x}_{m+1})\beta(m+1,n^i), \tag{2.32}$$

where the possible transitions from state n are to states $n^1, \cdots, n^k$. (For computational efficiency, the values of $b_n(\mathbf{x}_m)\beta(m,n)$ should be computed and stored, since these are used both in the recursion (2.32) and in the formula (2.30) for transition statistics.)

There is experimental evidence to suggest that the Viterbi algorithm, though theoretically inferior, since it does not provide maximum-likelihood estimates (when applied to training) or the overall likelihood of a word given a model (when applied to recognition), is as good as the forward-backward algorithm in practice, at least for some applications [103,104,117].

A Bayesian estimation procedure for estimating emission probabilities, assuming the availability of transition probability estimates, is described in [154]. This also involves iterative application of the Viterbi algorithm to training data, but uses speaker-independent prior probability distribution estimates for the model parameters, whose inclusion was found to improve the recognition performance of the models generated. It could be modified to incorporate reestimation of the transition probabilities.

A maximum mutual information estimation procedure, differing from the usual maximum likelihood estimation in that all the models' parameters are estimated together to optimise the discrimination between different words, has been observed to yield improved recognition performance [115].

There are other algorithms, including some based on Lagrangian techniques, which can be used to estimate both emission and transition probabilities [103,104].

A problem with many HMM training techniques is that they find locally optimal values in the space of model parameters which are not guaranteed to be globally optimal. Improvements in this respect have been obtained (at consider-

able computational expense) using a simulated annealing process [108], in which random perturbations are allowed to shake the parameter values out of a local optimum so that the globally optimal values can be found. Another approach [117] is to improve the chance of reaching the global optimum by deriving a good set of initial parameter values from training data by a procedure involving word segmentation.

The advantages of the HMM approach are that more of the information contained in several training tokens can be incorporated in a single word model than could be incorporated in a single template, and that probabilistic information from frequency-of-occurrence statistics can be conveniently represented; but to make use of these advantages several training tokens of each word are required. The greater the number of parameters to be estimated, the greater the quantity of training data required to make the estimates reliable [117]. One HMM per word has been found to give fairly good results in speaker-independent word recognition, though not always as good as those obtained (with considerably more computation) using multiple templates [103,104].

Improvements have been found to result from using two HMMs per word (one for male and one for female speakers, or by a clustering procedure), instead of pooling all the training data to make a single model for each word [107,110,116].

An adaptation procedure for HMMs, based on the forward-backward algorithm, has been implemented [253]; this allows models to be improved as more examples of the words become available, by a weighted averaging of the original training statistics and those derived from the new input, and can be used to adapt the models to a new speaker.

2.5.2: Modifications to improve duration modelling

A shortcoming of Markov models for representing spoken words is the assumption that the state transition and emission probabilities at each frame of the word are independent of the state transitions that have occurred earlier in the utterance and the frame vectors that have been emitted at earlier times. In particular, the probability that the process will remain in a given state at frame m, given that it is in that state at frame m-1, is taken to be independent of the number of frames preceding frame m-1 during which the process has been in that state. In the common and computationally convenient case where each state has a transition to itself to allow indefinite repetition, this gives rise to an exponential probability distribution for the duration of each state, which does not correspond well to the characteristics of real speech [106,107,119]. Modifications of HMM-based recognition have been proposed to overcome this problem, by incorporating durational information into the dynamic programming procedure [106,107,110,119], or by adjusting the word distances after alignment to take account of duration probabilities [107,110]. (Hidden semi-Markov models [109,119], which incorporate a specification of the probability distribution of the duration of each state, can be trained by an extension of the forward-backward algorithm [109].) Improvements in recognition accuracy have been observed to result in some cases [106,107,110]. It is also possible to improve state duration modelling, while retaining the computational simplicity afforded by the first-order Markov property, by using multiple "copies" of each state (i.e. states with identical emission probabilities) in some suitably devised configuration [118,119].

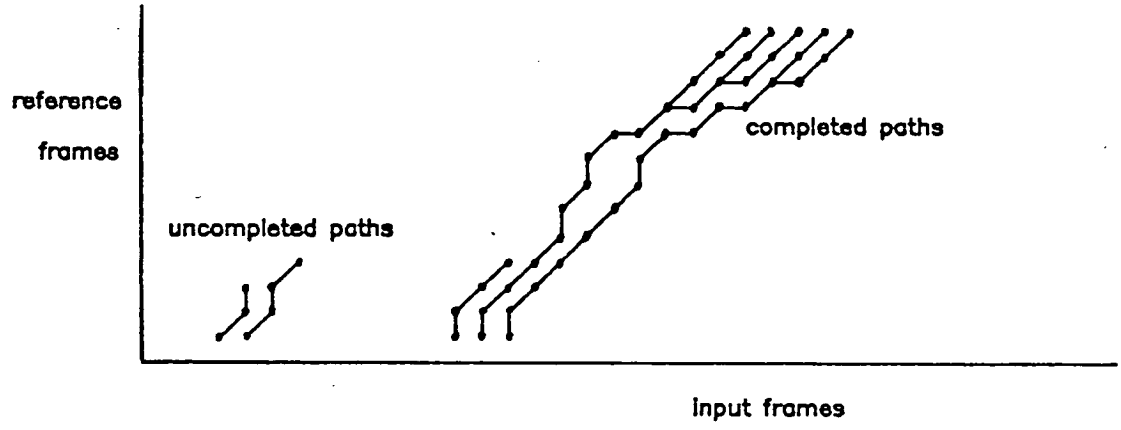
2.6: DTW applied to word spotting

Usually in a speech recognition system the aim is to recognise all the words spoken. The word spotting task is an exception to this: here the aim is just to find all occurrences of designated key words in a sample of connected speech, and there is no attempt to identify the other words in the utterance. This has applications in selection of those parts of a large corpus of speech (consisting for instance of intercepted radio messages) which are of interest for some specific purpose.

A method of matching a key word template to appropriate portions of a sample of connected speech has been described [130]. Like the isolated word recognition procedures already described, this involves time warping and frame distances; but in other respects there are differences due to the different nature of the problem. Instead of being given input with a specified word-beginning frame or region, the procedure must investigate each input frame to see whether an instance of the key word begins there. When a likely key-word-beginning is found, the end of the word is still undetermined, and so no final input frame can be specified to constrain the warping path. The decision required as output from the template matching process is not which word or sequence of words best matches the input pattern, but rather whether any parts of the input pattern are instances of the key word, and, if so, which parts.

The method described depends on defining a local similarity function F which is a weighted sum of frame similarity values at recent points along a partial path, with the weights decaying exponentially away from the current point. (The procedure does not give an overall distance or similarity measure for a completed path.)

Figure 2.7: application of DTW to word spotting



The search for a region matching the key word is illustrated in figure 2.7. From each input frame, a path is started, and is extended as long as the values taken by F continue to be above a set threshold. A path which is completed subject to this condition is taken as indicating an occurrence of the key word. There may well be several completed paths very close together, perhaps with points in common. In such a case a rule can be applied to ensure that the word is recognised as occurring only once there.

A modified version [132] of this word spotting method involves the use of multiple templates to form a composite template during the matching procedure. For each key word, several templates are stored, perhaps from different speakers; then, for each point (m,n) on a warping path, the frame similarity used is the maximum of the similarities between the mth input frame and the nth frames of all the templates. Thus the template effectively used for a path has its nth frame chosen from the nth frames of the available templates so as to maximise the similarity at the appropriate point on the path.

The threshold on F must be chosen to achieve a satisfactory rate of spotting of genuine occurrences of key words without introducing too many false alarms. It is best to choose the threshold value for each speaker and each key word individually; a problem with this is that the input speakers may not be available for training, and so some method of automatic adjustment to the characteristics of speakers is desirable [134].

A secondary testing procedure can be applied [135] for identification of the better "putative hits", using template-specific linear combinations of various matching statistics, to add to the information obtained in the primary matching process. A weight is assigned to each combination of a frame matching statistic and a template frame, depending on how significant that statistic is at that frame of the template for discrimination between true hits (on genuine occurrences of key words) and false alarms.

A word spotting technique based on segmentation of each template, with matching by a dynamic programming algorithm incorporating limits on segment duration, has been found to yield better results than the whole-word matching method with standard DTW [136].

2.7: Dynamic programming algorithms for connected word recognition

In a connected word recognition system, the input consists of words spoken without gaps between them. (Connected speech is not necessarily fluent speech: the latter has more coarticulation and assimilation between words, whereby the realisations of the initial and final parts of a word are affected by the preceding and following words [15], making the application of a template-matching system rather difficult [228].) The whole utterance is typically a sentence or a string of digits. Thus the recognition involves not only deciding what each word is but

also finding where words begin and end, and, usually, deciding how many words the utterance contains. (A rather similar problem occurs in isolated word recognition if the gaps between words are not long enough to be distinguished, by their duration, from stop consonant silences within words. The "Quiktalk" algorithm described in [59] is designed to cope with this problem, and is similar in principle to the two-level algorithm for connected word recognition described below.) A DTW system with word templates can make all these decisions together. Various DTW algorithms for connected word recognition have been devised; these are described in the subsections below.

Except where stated otherwise, the asymmetric scheme of weighting in the input direction (scheme (c)) is employed in these algorithms: this allows comparison (without normalisation) of accumulated distances obtained by matching different numbers of templates, or templates of different lengths, to the same section of the input speech.

2.7.1: The two-level algorithm

A two-level algorithm has been proposed [138] in which, in the first stage, each word template is matched against every part of the input, and then, in the second stage, these matchings are combined to give a recognition of a whole string of words.

In the first part of the two-level algorithm, word level matching, a template is chosen for each permissible input section (permissible in the sense that it is of such a length that at least one of the templates can be matched to it) to minimise the word distance obtained by matching against that part of the input. The output of the word level matching consists of a template index and a corresponding word distance for each permissible pair of starting and ending frames.

The second part of the algorithm is phrase level matching. This uses the information given by the word level matching and builds up the word reference string by a dynamic programming search in the (l,m) plane, where l is the number of the word (counting from the start of the utterance) and m is the input frame number. The accumulated distance $G(l,m)$ at each point is obtained by minimising over all possible beginning frames for an lth word ending at frame m. A backpointer, which is a record of the beginning frame used (or, equivalently, of the previous word's ending frame), is kept at each point (l,m) during the phrase level matching. The path ends at (L,M), where L is the number of words in the utterance and M is the number of input frames. If the number of words in the utterance has been specified, L is fixed and so the endpoint (L,M) is automatically determined; if not, several paths may be completed, with different values of L at their endpoints, corresponding to matchings of concatenations of different numbers of reference words to the input. In this latter case, L is chosen so as to minimise the total distance $G(L,M)$. The recognised sequence of words can be recovered (in reverse order) by tracing through the array of backpointers.

Two implementations of this two-level matching procedure have been given [138]: one suitable for computer simulation, in which tables of frame distances for all possible input-reference frame pairs are computed and referred to in the course of the word level matching, and the phrase level matching is not begun until the word level matching has been completed; and one suitable for real-time recognition, in which, as each input frame is read in, the word level and phrase level matching procedures are advanced to incorporate words and partial phrases ending at that input frame. In a real-time implementation, it is possible to avoid repeated computation of frame distances (during word level matching operations with the same template but different starting frames) by temporarily

storing all the required frame distances for the current input frame.

An algorithm similar to the two-level algorithm but allowing an overlap or a gap between successive words has been formulated and tested [167]. The best results (with templates formed from connected speech as well as from isolated words) were obtained when no overlap was allowed but a gap of up to 15 or 20 frames (150 or 200ms) was permitted.

2.7.2: The sampling algorithm

An algorithm which reduces the computation involved by matching templates only to selected parts of the input is described in [139,140,141,142]. This builds up strings of words from left to right, and at the lth stage keeps a list of the best few candidate strings of identifications of the first l words, with their ending frames and accumulated distances. To extend a partial recognition string, templates are matched to the portion of input speech following its ending frame; the UELM search area constraints (described in section 2.3.6) are used. An overlap of up to Δ frames between successive words is allowed and, within the range defined by this parameter, every Ith input frame is tried as a starting frame. A string is considered complete when its last word ends within a specified number of frames of the end of the input pattern.

Once all strings of words have been either completed or abandoned, the one with the smallest overall distance can be taken as the recognition of the utterance, or else a post-error correction procedure can be employed. In the post-error correction, a whole-utterance template is constructed for each of the best few complete strings already found, by concatenating the templates in the string, and is matched against the input utterance by a large-scale constrained-endpoint DTW; the final decision as to which string best matches the input can

then be based either purely on the distance obtained in this way or on the average of this and the original distance derived in building up the string.

A modified version of the sampling algorithm, with reduced weighting near word boundaries where coarticulation occurs, is described in [147].

Unlike the other connected word DTW algorithms described here, the sampling algorithm does not guarantee optimality of the sequence of words found.

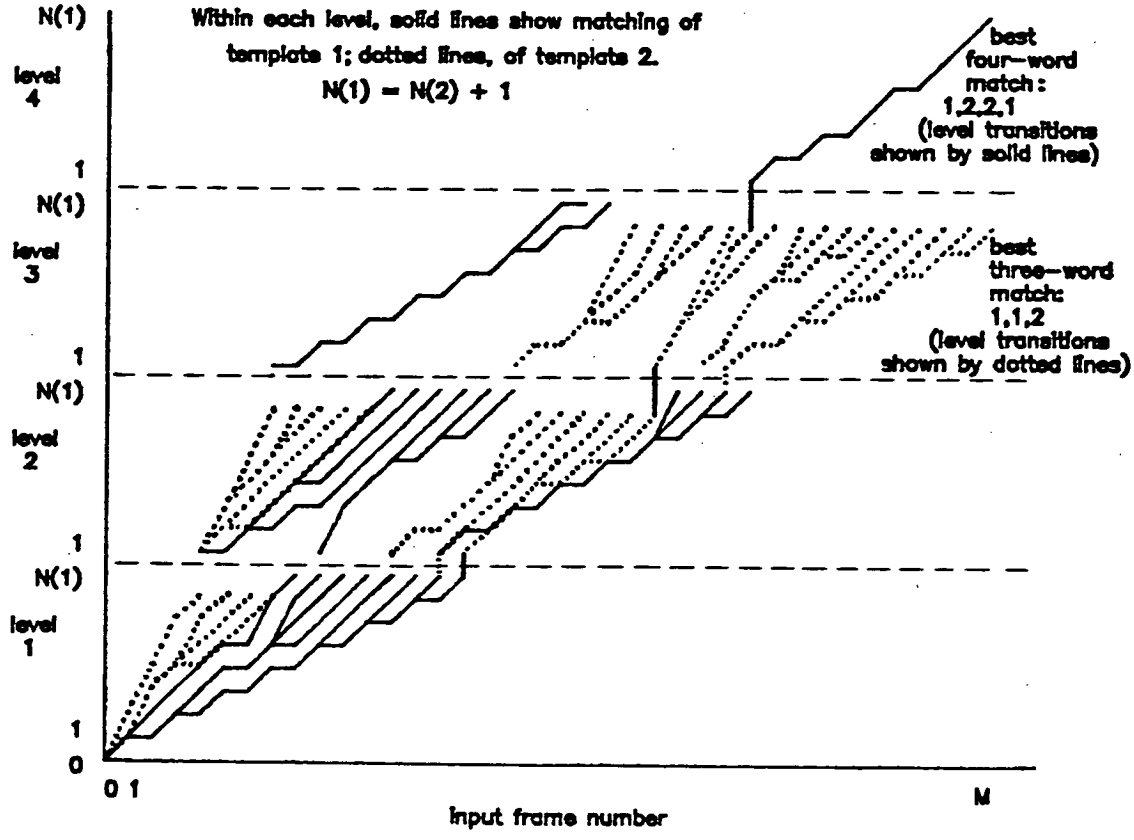
2.7.3: The level building algorithm

The level building algorithm [143] is proposed as a more computationally efficient implementation of the basic idea of the two-level algorithm. The fundamental difference between the two is that, whereas the two-level algorithm minimises accumulated distance at each possible partial phrase ending frame m firstly over template index v (separately for each possible beginning frame m' of a word ending at m) and then over word beginning frame m', the level building algorithm minimises it firstly over m' (separately for each v) and then over v. (Again the minimisation is done separately for each possible value of l corresponding to the ending frame m.) The meaning of "level" in this algorithm is different from that in the two-level algorithm: here there is no division into word level and phrase level matching, and a "level" is a value of l. The improvement in efficiency is because duplication of accumulated distance calculations occurs only due to matching the same word at different levels in the same region of the input pattern: there is no duplication for different beginning frames of the same word at the same level, since all the matching for a given word and level is carried out in a single application of the basic DTW procedure.

The operation of the level building algorithm is illustrated in figure 2.8. The computation proceeds level by level, and, at each level, template by

template. At the beginning of each level, there are various possible previous frames, each with its accumulated distance derived from the previous levels. (In the case of the first level, there is only one previous frame, namely input frame 0 (added for convenience of notation), with accumulated distance 0.) In the

Figure 2.8: the level building algorithm for connected word recognition



notation used above for the two-level algorithm, the accumulated distance to input frame m at the beginning of level l is $G(l-1,m)$. For each template, a DTW matching is carried out starting from these previous-level ending frames and their accumulated distances, which gives a set of ending frames for that template on the current level, with corresponding accumulated distances. (The optimal starting frame of the word does not have to be found separately for each ending frame: paths to all the ending frames are found by the single DTW operation.) Once all the templates have been matched at the lth level, accumulated distances $G(l,m)$ are found by minimising for each ending frame m over the accumulated distances calculated there using the various templates. As usual, a record must be kept at each point of how that point has been reached, so that the words can be recovered at the end of the whole process.

Within each level, paths may be allowed to start within a specified interval at the beginning of a template, and to end within a specified interval at the end of it, to allow for the reduced pronunciations which often occur in connected speech [143,144,145,146]. This modification has been found to be important for the attainment of optimal performance, when the templates are derived from isolated-word utterances [145].

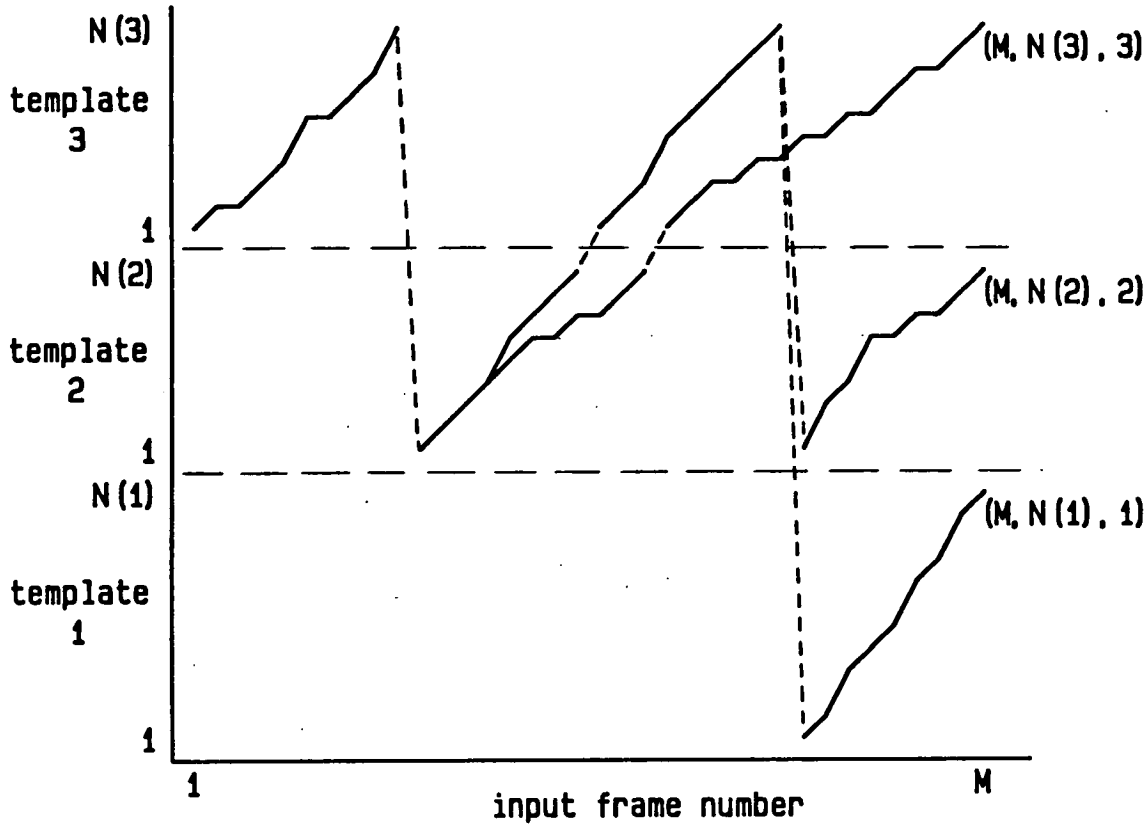
A reduced level building algorithm [143] incorporates a restriction of the lth-level ending frame $M(l)$ to a set range around the value m which minimises the length-normalised accumulated distance $\frac{G(l,m)}{m}$, and an accumulated distance threshold to eliminate badly-matching templates at each level (basically as described in section 2.4). This reduced algorithm involves about the same number of frame distance computations as the sampling algorithm, but yields significantly better recognition rates [144,145].

2.7.4: The one-stage algorithm

A conceptually simple connected word recognition algorithm, originally formulated by Vintsyuk in 1971 [3], is described in [148,162]. Essentially the same algorithm has been adopted in various connected word recognition systems [151,153,156,157,159,160,185]. It has the advantage over the algorithms described above that only one accumulated distance calculation is performed for each input-reference frame pair. It also requires only a fairly small amount of storage for data obtained during its execution.

The principle of its operation is illustrated in figure 2.9. For the purpose of the dynamic programming procedure, the word templates are made into a compound template, with special rules for going from the end of one word template to the beginning of another. A frame in this compound template has a word template index v, and a frame number n within the word template; so a point in the input-reference plane can be specified by three coordinates (m,n,v), where as usual m is the input frame. Where $n > 1$ (or 2, if path constraints such as those specified in section 2.3.3 are used), the permitted predecessors of (m,n,v) are as specified by the usual path constraints, except for the extra coordinate v. Where $n = 1$ (or 2, depending on the path constraints), the preceding point may be $(m-1,N(v'),v')$ for any template index v' (or it may be any preceding point in template v permitted by the ordinary path constraints, such as $(m-1,1,v)$). (Here $N(v)$ is the number of frames in the vth word template $R(v)$.) Apart from this modification to the local path constraints, the DTW proceeds as for the isolated word case, one input frame at a time. One or more columns of accumulated distances, depending on the path constraints, must be stored at each stage in the procedure. (As is usual in a connected word system, a record must also be kept of the word templates used on the path to each point reached, and this is

Figure 2.9: the one-stage connected word recognition algorithm



accomplished efficiently by means of arrays of backpointers.) A complete path may end at the final frame of any of the word templates, that is, at any point $(M,N(v),v)$; this may be relaxed to allow for errors in identifying endpoints. Thus there will generally be at least V complete paths formed, where V is the number of words in the vocabulary; a decision is made among the paths on the basis of their overall distances.

It is possible to make allowance for coarticulation and reduction at word boundaries by permitting jumps to $(m,1,v)$ not just from $(m-1,N(v'),v')$ but also from $(m-1,n,v')$ where $N(v')$ - n < some constant. (This allows for the omission of the end of a word in the input speech, which is more usual in natural speech than the omission of the beginning of a word. But a similar modification could be applied to the beginnings of word templates also.) Further modifications [148,162,163] are the inclusion among the word templates of a silence or noise template (to allow for pauses) and of a pseudotemplate with a fixed distance from all possible frame representations (to match input which does not match any of the reference words well). Each of these has just one frame.

Implementations have been described [160] with weights other than scheme (c); the total weight on a path through a word template depends in this case on the length of the template as well as on the number of input frames matched to it, but a normalisation for template length is carried out whenever a path reaches the end of a template, so that the total weight after normalisation is the same for all sequences of templates ending at a given input frame. (Similar normalisation could be applied in any of the other algorithms described in the preceding three sections, to allow the use of various weighting schemes within each matching of a reference word.) Type I local path constraints and a symmetric weighting scheme were found to be better than Itakura constraints and the asymmetric scheme. A modification has also been described [153] in which different weights are assigned to different frames of a template (cf. the frame-specific weighting suggested in [130] and mentioned in section 2.3.4 above), with a similar normalisation for total weight at the end of each word. A connected word recognition algorithm for use with variable frame rate analysis, incorporating weighting according to the durations of the frames being matched together, has been implemented, and was found to give better results than without use of

duration information; the variable frame rate analysis technique (rather similar to trace segmentation with averaging) improved the recognition performance while reducing the amount of computation [168].

One disadvantage of the one-stage algorithm in its basic form (relative to the two-level and level building algorithms) is that it includes no specification of the number of words in the utterance. Such a specification could, however, be incorporated by means of syntactic constraints [148], as described below.

Implementations of the one-stage algorithm designed to reduce the number of memory access operations required are described in [169].

2.7.5: Incorporating syntactic constraints

In a connected word recogniser with a large vocabulary, it is desirable to be able to make use of any syntax known to apply to the input utterance, in order to reduce the number of words that have to be tried at each point, and to increase the recognition accuracy by ruling out likely-sounding but impossible word sequences. (The use of syntactic information has been found to result in greatly improved recognition accuracy in a system with input of sentences spoken as isolated words [12,126,127].) All the above algorithms can be modified fairly easily to incorporate syntactic constraints.

In the sampling algorithm, the matching of templates to extend partial recognition strings can be restricted so that only syntactically permissible words are tried.

In the level building algorithm, levels can be replaced by states in a loop-free transition network representing a simple grammar, and adding 1 to the level number l to get to the next level when passing on accumulated distances can be replaced by making a transition to another state [143]. (The record kept

of the path taken to each point will have to include information about which state transitions have been made.) This adaptation of the level building algorithm is described in more detail in [150]. By arranging the computation so that all levels are extended in parallel along the input, it is possible to dispense with the requirement that the syntax be loop-free, since in this case each transition need not always be from a preceding level.

The two-level algorithm could be modified [145] to allow the application of syntax, specified again by a transition network (not necessarily loop-free). In this case, the phrase level matching would have syntactic states in place of word numbers, and for each input section the optimal template and distance would have to be determined (in the word level matching) not just from the whole vocabulary but from each subset of the vocabulary corresponding to a syntactic state transition.

An advantage of the two-level algorithm [145] is that it can generate the K best candidate strings, for any specified value of K – which can be useful in an application where the direct implementation of word sequence constraints is unwieldy (as in the recognition of spellings of names from a directory [130]). This is done by recording the K best words and distances for each input section (or all the word distances if $K > V$), and retaining the K best partial strings of words at each point in the phrase level matching.

In the one-stage algorithm, each reference word can be given a "from predecessor" set, which is the set of all words which may precede it; then the preceding points allowed for $(m,1,v)$ will include $(m-1,N(v'),v')$ only for those template indices v' for which $R(v')$ is in $R(v)$'s "from predecessor" set [162]. The points at which warping paths may start will be $(1,1,v)$ only for those values of v for which $R(v)$ is designated as a word which may begin an utterance; similarly, paths will be allowed to end at points $(M,N(v),v)$ only where the words $R(v)$ are

permissible final words. If $R(v)$ occurs at more than one position in the syntax, then several copies of $R(v)$ can be included in the compound template, each with a different "from predecessor" set; they will be treated as different words during the DTW procedure, except that the frame distances can be computed just once (for each input frame) and used for all the copies. The syntactic network need not be loop-free. (A particular case, where in fact the network is loop-free, is where the syntax specifies only the number, L, of words in the utterance; then there are L copies of each template, the "from predecessor" set of the lth copy of a template consists of the $(l-1)$th copies of all the templates, the permissible initial words are the first copies of all templates and the final words are the Lth copies. The algorithm with this syntax is identical to the level building algorithm.) Details of the implementation of the syntactically constrained one-stage algorithm are given in [148] and [185].

If a Markov modelling approach (section 2.5) is adopted, an integrated syntactic and acoustic network can be constructed [159,196], in which each word in the syntactic network is replaced by the transition network for the Markov model of the word (with possible insertion of silence states at word boundaries). Then the Viterbi algorithm can be applied to this integrated network just as it would be applied to the transition network for a single word. This connected word recognition procedure corresponds to the one-stage algorithm with syntactic constraints in the same way that the isolated word Viterbi algorithm corresponds to ordinary isolated word DTW. (Other connected word recognition procedures using HMMs include one using stack decoding [6,8,10], and one using level building with Viterbi matching for each individual word [165]. The first of these, unlike procedures using the Viterbi algorithm, computes an overall likelihood for the recognised word sequence, by storing and adding the probabilities for all alignments of the sequence of word models with the input.) Probabilistic

information on the occurrence of words and word sequences can conveniently be incorporated into a Markov-model-based recogniser [8].

2.7.6: Training for connected word recognition

Because words often have different characteristics when spoken connectedly (due to coarticulation with preceding and following words), training on isolated words may not be satisfactory for a connected word recognition system. To improve performance in the presence of interword coarticulation, a technique of training on words from connected speech may be used. Training tokens extracted from connected speech can be used to construct templates corresponding to different speaking rates and degrees of coarticulation, which can then be used alongside the templates obtained from isolated word tokens [149,158]. In an HMM-based recogniser, strings of words may be used for training, instead of isolated words [154]. It is also possible in a template-based recogniser to improve templates trained initially on isolated or manually-extracted words, by matching concatenated templates to known connected strings of words to find the word boundaries, and so extracting instances of the words for further training [166,167,170].

2.8: Summary and discussion

In the preceding sections, a particular class of word-based automatic speech recognition techniques has been described in some detail. These techniques are characterised by their use of dynamic programming algorithms for time alignment of the input speech with word reference patterns. Two main types of reference patterns have been considered, with their respective forms of dynamic programming algorithms: templates (the simpler of the two types), for which the

dynamic programming algorithm takes the form known as dynamic time warping (DTW), and the more sophisticated hidden Markov models, to which the Viterbi algorithm can be applied. The main emphasis in the discussion of these techniques has been on template matching and DTW; but the conceptual link between this approach and the hidden Markov model approach using the Viterbi algorithm has been demonstrated.

It is evident from published results [19,49,76,82,101] that dynamic programming is an effective method of performing time alignment of speech patterns, and that DTW provides a substantial improvement over linear time registration in template-based word recognition, as measured by recognition rates. (Some methods of word matching without non-linear time alignment have, however, yielded good results. These include methods using segment-specific VQ codebooks (without fine temporal information) [123,124], or histograms of occurrences of VQ labels [125], in equal-duration segments, for which linear alignment is adequate because the temporal variations are represented implicitly in the histograms or codebooks. These methods require several training utterances for each word, and are thus most applicable to speaker-independent recognition.)

The major disadvantage of DTW, especially for large vocabularies, is the large amount of computation required, particularly for the numerous frame distance calculations. (Other methods of non-linear time alignment have been devised [82,122] which require less computation, but the recognition accuracies of these methods are inferior to those obtained with DTW.) Therefore a good deal of effort has been devoted to finding ways to reduce the computational load without losing too much recognition accuracy.

Among the most promising reduction techniques are preliminary trace segmentation [76,80,81] (combined with search area restrictions [76,81]); accumulated distance thresholds [18,54], progressive rejection procedures [47] and beam searching [64,66], for rejection of templates (and, in the case of beam searching, individual paths) before completion of the matching; and, for large vocabularies in particular, match limiting by an initial simple comparison [229,230,231,232,233,235,236]. (Note, however, that the first and last of these may not allow the DTW matching to start until the input word has been completed.) These three techniques could be combined quite effectively in an isolated word recogniser. (Progressive rejection procedures and match limiting have in common the principle that unpromising candidate templates need not be matched to the input in full. This will be discussed further in chapter 3. The ideas of segmentation and of match limiting are integrated together in the multiple-stage recognition system which will be described in chapter 4.) Beam searching is also applicable to connected word recognition using the one-stage or Viterbi algorithm, and has frequently been so applied [8,11,148,152,159,196].

The variable-length trace segmentation described in [80,81] has some of the merits of the fixed-length trace segmentation procedure, and allows the matching to begin during the speaking of the input word; but it does not result in normalisation to a constant number of frames per word (and hence the severe search area restrictions of [76] cannot be applied). This technique, unlike fixed-length trace segmentation, can be applied to connected word recognition with an unknown number of words in the input utterance [147,152,161]. Numerous acoustic segmentation techniques have been devised [52,66,69,76,80,81,82,92,147,152,161], varying in the criteria for defining segments, the derivation of a representation for each segment (or each segment boundary) and the use, if any, made of information on segment durations.

Vector quantisation (VQ) [32,78,80,103,104,156,157,159,160,171] can reduce the amount of computation to be done in evaluating frame distances. There tends to be a loss of accuracy in recognition when VQ is applied; this can be counteracted by using a large quantisation codebook, but then significant computational savings are achieved only when the vocabulary is fairly large.

One motive for applying the above computation reduction techniques is the desirability (for most applications) of real-time or near-real-time recognition. Another approach altogether (though the two can in some cases be combined) is to use special hardware on which DTW can be performed rapidly. Descriptions of devices designed specifically for DTW computation are given in [183,185,186,187,188,189,190,191,192,194,195]. DTW algorithms lend themselves to parallel processing; several parallel processing schemes for DTW are described and compared in [184], and the arrays of [183], [187] and [192] implement two of these schemes. Systolic array architectures for connected word recognition, using algorithms similar to the two-level algorithm, are described in [195]. A pipelined structure for beam searching is described in [193].

The choice of local path constraints and weighting functions in the basic DTW algorithm tends not to make very much difference to the quality of recognition obtained; however, the results of [49,60,63,140,142] suggest that Itakura or Type II constraints (in the notation of [60], as defined in section 2.3.3 above), with weighting according to movement in the input direction, give a slight advantage in recognition accuracy over other combinations while being quite economical computationally. Where paths are confined to a narrow band from the initial to the final point of the warp, the ban on repetition of the step (1,0) in the Itakura constraints has been found to be unhelpful [68].

The superiority of Itakura constraints to Type III [63] might seem surprising, since the former constraints yield only an approximation to the optimal alignment given by the true dynamic programming formulation of the latter. Presumably the explanation is that the word distance computed using the Itakura constraints tends to be more suboptimal for incorrect-word matches than for correct-word ones. This can be understood by considering the nature of the cases where an optimal path, obtained with Type III constraints, is liable to be excluded by the condition for horizontal steps in the Itakura constraints. Such cases include those where one frame of the template (assumed to be on the vertical axis) is similar to several successive input frames but the adjacent frames of the template are dissimilar: horizontal steps will tend to occur at that template frame during the search for the alignment path, and so a horizontal step in the minimal-distance path may well be forbidden because of a previously chosen horizontal step, one input frame earlier, at the same frame in the template. Such instances of isolated well-matching template frames seem more likely to occur in the alignment of incorrect-word templates than with correct-word templates.

Results of experiments on endpoint relaxation indicate that in some cases it can be helpful [51,58,63] but in others it increases the error rate [30,91]. Methods of adjusting the endpoints by preliminary testing [58,91] or methods using extended input and initial and final silence frames appended to the templates [91] (cf. the noise template method of [160]) appear to be better than simple endpoint relaxation. Where a symmetric weighting scheme is adopted, however, an endpoint relaxation technique such as that adopted in edge-free staggered array DP [38,73] can attain a similar effect to the silence-frame technique, without the need for silence frames (but with extended input and reference patterns). (The appropriateness of any endpoint modification technique will depend on the characteristics of the method used to locate the endpoints in the first

place, and on the conditions in which the recogniser is being used.)

Any speech recogniser is only as good as its reference patterns. For speaker-dependent recognition, a robust training procedure [30,31], which obtains two sufficiently similar repetitions of each word and averages them, can help to ensure that the templates formed are free from mispronunciations and extraneous noise. For speaker-independent recognition, a clustering analysis [48,53,54,55,100] – or, alternatively, editing and condensing [102] – of training data from a representative set of speakers is an appropriate means of deriving templates.

A development of the idea of training to the speaker is progressive adaptation of templates during the input of speech to be recognised. This will be discussed in chapter 3, and a description of a recognition system incorporating template adaptation, and results obtained with this system, will be given in chapters 5-7.

Recognisers using whole word templates tend to perform badly on vocabularies containing words which differ only in short sections. Refinements to the basic template-matching method, such as discriminative networks [84,85] to eliminate the effects of linguistically insignificant differences, and two-pass decision procedures using weighting to emphasise distinguishing features [61,75], result in improved performance on such vocabularies. In the special case where durational information is important for distinguishing words, the techniques of [70,83] can be helpful. In each case, DTW is still used, with appropriate modifications. The maximum mutual information training procedure for HMMs [115] is similar to the discriminative weighting technique, in that it takes into account characteristics of the vocabulary as a whole, rather than only of each word individually.

For recognition of connected speech, the one-stage algorithm [148,151,153,162] has certain advantages over the other algorithms described: it performs each accumulated distance calculation only once, and it can easily be adapted to incorporate syntactic constraints. If the number of words in each input utterance is fixed, however, the level building algorithm [143,144] may be better [190], since it always finds the best-matching string of L words for a specified value of L. (The one-stage algorithm can be converted into an implementation of the level building algorithm by imposing appropriate constraints.)

The use of templates extracted from connected speech can significantly improve the performance of a connected word recogniser [149,166]. It is also helpful to permit the omission of the end (or beginning) of a word, which commonly occurs in connected speech [145]. However, the word template matching approach has the fundamental limitation that, because the basic unit is the word (rather than something smaller), it is difficult to make full allowance for coarticulation effects by incorporating detailed phonological rules. Thus it may be better to use a phoneme-based (or phonetic-segment-based) strategy where natural continuous speech is to be recognised [15]. The "phonemes" used in a recognition system will not necessarily be what would be called phonemes in standard linguistics; it may well be found necessary [237] to include several realisations of each (strictly so-called) phoneme. (Dynamic programming algorithms, similar in principle to the connected word recognition algorithms described in section 2.7, have been formulated to match continuous speech with concatenated templates or models for subword units [171,173,174,176,178,179,180], and to compare a phoneme lattice derived from unknown input speech with reference lattices or models [182]; related algorithms can also be used in automatic training of a recogniser on continuous speech [172,177,181].)

Another disadvantage of the word template matching approach is the amount of training required for each new speaker (in a speaker-trained system) or for each new word added to the vocabulary (in a speaker-independent system using clustering analysis for template creation). This too is less of a problem in a system based on units smaller than words, since there are fewer reference patterns to be determined, and also partial-vocabulary training (as in [237]) is more feasible.

Even in a system where the primary recognition strategy is one of segmentation into phoneme-sized (or smaller) units, or of matching with phoneme templates, DTW matching of word templates may have a part to play: as suggested in [137], it can be used to verify the recognition hypotheses produced by the primary analysis. (The verifier described in [137] used templates generated by a synthesis-by-rule program, but it would be quite possible to use natural word templates instead.) For verification in a continuous speech recogniser, an algorithm such as ZIP [155] could be used, with either concatenated word templates or synthesised speech on the reference axis. (The post-error correction procedure of [141] is a verification procedure using concatenated word templates.)

Dynamic programming has been successfully applied to speech recognition using hidden Markov models [103,104,154,156,159,171,196]. (While template matching using DTW is technically a form of hidden Markov model matching by the Viterbi algorithm, in practice there have tended to be two distinguishable approaches.) The Markov modelling approach has the advantage that more information about the variability of the pronunciation of a word can be incorporated into a Markov model than into a template. The use of Markov models instead of templates has been found to lead to improved accuracy in recognition [156], or to similar accuracy with reduced computation [103,104,159]. However, a large amount of training is required for a Markov modelling system [103,104],

which may make it unsuitable for some applications. The case for progressive adaptation during recognition sessions is perhaps even stronger for Markov model systems than for template-based systems, since the asymptotic optimal performance is reached more slowly as the amount of training data increases. A method which could be applied to accomplish such adaptation has been described, and has been found to allow improvement of models [253].

Word recognition systems using dynamic programming techniques have been produced commercially by various manufacturers [198,203,204,216]. Many of the available recognisers are designed for isolated word recognition only, but others, such as the NEC DP-100 [196,207] and DP-200, the British-produced Logos [185] and Marconi SR128 and the Verbex 1800 [207], incorporate connected word algorithms. A few of the commercially produced recognisers, such as the Verbex 1800 and 4000, and Dragon speech recognition software [206], use Markov models; IBM has produced an experimental HMM-based recogniser with a 5000-word vocabulary, designed for dictation of sentences spoken as sequences of isolated words [172,175]. Comparative evaluations of speech recognisers in terms of accuracy and other features are given in [197,198,200,202,207]; a comparison of human and machine speech recognition is reported in [71]. Standards for evaluating recognition systems are in the process of development [199,201,205].

An important aspect of many potential applications of isolated and connected word recognition is that the possible sequences of words are strongly constrained by the syntax, semantics and pragmatics of the task [12,16]. The use of syntactic constraints to restrict the output of the recognition system can greatly improve its accuracy, whether the input consists of a sequence of isolated words [12,126,127] or of a connected utterance [16,165]. In a dialogue system, the use of a semantic model to exploit the relations between successive sentences can

further improve the performance by allowing acoustic recognition errors to be corrected automatically, or detected and corrected by querying the user [12,16,129]; and where understanding rather than transcription is the goal some word errors may not affect the outcome [129]. Thus the task may be accomplished with much greater reliability than would be possible using the word-level acoustic pattern-matching alone.

The above discussion indicates that, although so much research has already been done on word-based speech recognition using dynamic programming algorithms, there are still areas where further investigations could be worthwhile. These include the formation of templates, networks or statistical models (in an initial training process or by adaptation), and the construction of effective matching procedures and decision rules to be used with them, to provide optimal discrimination among the words of a specific vocabulary (rather than simply optimal modelling of each word individually) [61,75,84,85,102,115,254,256]; and the design of whole speech systems, incorporating use of the available information at all levels [12] and appropriate interaction between the machine and the user [12,220,222,223,224,225,226,227,228]. Some specific topics for investigation are mapped out in chapter 3.

CHAPTER 3

AREAS FOR FURTHER RESEARCH AND IMPROVEMENT IN WORD-BASED SPEECH RECOGNITION

3: AREAS FOR FURTHER RESEARCH AND IMPROVEMENT IN WORD-BASED SPEECH RECOGNITION

3.1: Introduction

For the reasons discussed in chapter 1, techniques using whole-word reference patterns have been adopted in many experimental and commercial speech recognition systems; and it seems likely that such word-based techniques will continue to be the most effective methods available for many practical applications of automatic speech recognition. Approaches based on units smaller than the word – particularly using phoneme-level Markov models – are becoming increasingly practicable [216], and these will partly replace the word-based techniques, especially for large-vocabulary recognition (where the use of a few phonetic units can eliminate the training requirements imposed by word-based recognition) and for the recognition of continuous speech (where a phonetically-based approach should be able to cope better with coarticulation and related effects). However, there will probably still be applications, particularly in isolated word recognition for strictly defined tasks (where a small vocabulary is often adequate), in which word-based techniques are more appropriate.

Among the most effective word-based recognition techniques are those described in chapter 2, using templates and hidden Markov models to represent the words of the vocabulary, and incorporating dynamic programming algorithms (DTW and Viterbi) for optimal time-alignment.

It is therefore of practical as well as theoretical interest to investigate possible improvements to the existing template-matching and HMM-based word recognition techniques, and to discover the relative effectiveness of different options. While a great deal of research has been conducted with word recogni-

tion systems over the past few years, there are still questions to be answered and possible enhancements to be explored. It is of particular importance to study the effects of the interaction between system and user on the performance of (isolated or connected) word recognisers, since this interaction will occur, in some form, in any practical application of a speech recognition system.

Some of the outstanding questions to be answered, and areas for improvement and development, in the light of the results reviewed in chapter 2, are considered in the sections that follow. Experiments conducted to explore these areas will be described in later chapters.

3.2: Time segmentation and segment representation

One fairly simple topic for investigation is the comparison of the various forms of trace segmentation and similar techniques, applied as preprocessing for words to be compared by DTW, referred to in section 2.3.7 above.

Numerous acoustic segmentation techniques have been devised [52,66,69,76,80,81,82,92,147,152,161,168], varying in the criteria for defining segments, the derivation of a representation for each segment (or each segment boundary) and the use, if any, made of information on segment durations.

These techniques are useful in reducing the number of vectors representing each word (and hence the computation involved in comparing the words); in normalising each word to a standard length (though not all the techniques permit this); in representing more fully those parts of each word which show more rapid acoustic change (which are likely to be the parts with a higher density of linguistically significant information); and in performing some initial non-linear timescale adjustment.

The question of the different possible methods of deriving segment representations does not seem to have been addressed systematically by any of the previous researchers who have used such techniques. In some systems, a vector is derived by linear interpolation at each segment boundary [76,80,161]; in others, the nearest vector to each segment boundary is selected [81,82,147]; and in others, a vector is derived for each segment (rather than each boundary) by averaging of the vectors in the segment [152,168]. It is of interest to know which of these techniques yields the best performance in a recognition system, for any given segmentation of the input and reference words. Experiments to compare these segment representation techniques have been performed [97], and details of these experiments and their results are given in chapter 4 below.

Comparisons of trace segmentation with linear time segmentation (in both isolated and connected word recognition systems) [81,92,161] have shown that the non-linear adjustment provided by trace segmentation, with less severe compression in regions of rapid spectral change, yields an improvement in recognition performance over the equivalent linear segmentation. However, there are also some results – those of Ney [152] for connected word recognition – which show the same degree of degradation of recognition performance with trace segmentation as with linear downsampling of the data. There are several possible explanations for these results, since the vocabularies used were different (and indeed taken from different languages: French [81,161], English [92] and German [152]), and Ney's experiments, unlike those of the other researchers, used a cepstral representation and the averaging method of segment representation. The experiments reported in chapter 4 include a comparison of trace segmentation and linear segmentation in isolated word recognition for two English-language vocabularies, with mel cepstral coefficients [23] as the acoustic representation, and with different segment representation techniques.

3.3: Refinement of the recognition decision procedure

The basic DTW template-matching isolated word recogniser has a rather crude strategy for determining which template best matches the input word. It performs a computationally costly DTW matching operation for every template, even though some templates, in a typical vocabulary, will be very dissimilar to the input word and should not require such a sophisticated comparison to eliminate them from consideration. This exhaustive comparison approach is a very poor approximation to the human word recognition process, which seems to proceed by narrowing down the range of possible recognitions progressively as more of the input is received and processed [15].

There are various possible modifications of the recognition procedure (as described in section 2.4 above) which allow poorly-matching templates to be eliminated without full-scale DTW matching. Some of them do this by beginning the matching process for every template but abandoning some templates part-way through the comparison [12,18,47,54,64,66]; others use simplified comparison techniques to eliminate unpromising candidates without even starting the full DTW matching for them [230,231,232,233,235,236]. Features which have been used or proposed for a rapid preliminary comparison to exclude unlikely recognitions include word duration [229]; characteristics of the end of the word [230]; vector quantisation distortions obtained for word-specific codebooks [231,232,233]; averaged acoustic parameter vectors for coarse time segments, compared using linear alignment or simplified non-linear alignment [229,235]; Fourier coefficients of gross spectral features across the word [236]; broad phonetic string classifications [234]; and prosodic characteristics such as syllabification, stress and rhythm [15].

Whether the DTW matching process is begun for all words and then abandoned before completion for some of them, or whether a separate preliminary comparison stage is employed, the condition for abandoning templates can take any of three basic forms, illustrated in figures 3.1-3. The condition to abandon a template can be a threshold on a distance measure, not dependent on the distances for the other templates under consideration (figure 3.1) [12,54]; or a condition on distance relative to the best template's distance (figure 3.2) [64,66,232,233]; or one which ensures retention of only a preset number of templates (figure 3.3) [47,229,235,236]. (Conditions combining more than one of these three types can also be devised [232].) The first of these can result in the

Figure 3.1: elimination of templates using an absolute distance threshold

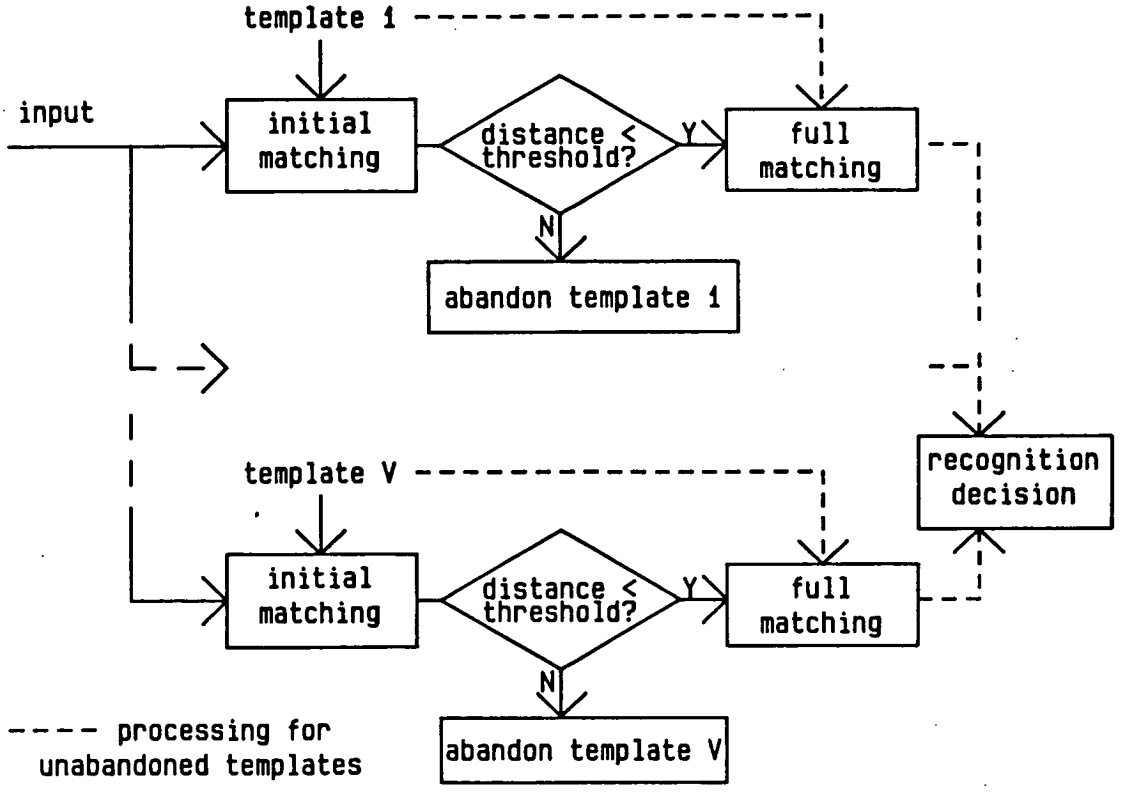


Figure 3.2: elimination of templates using a relative distance threshold

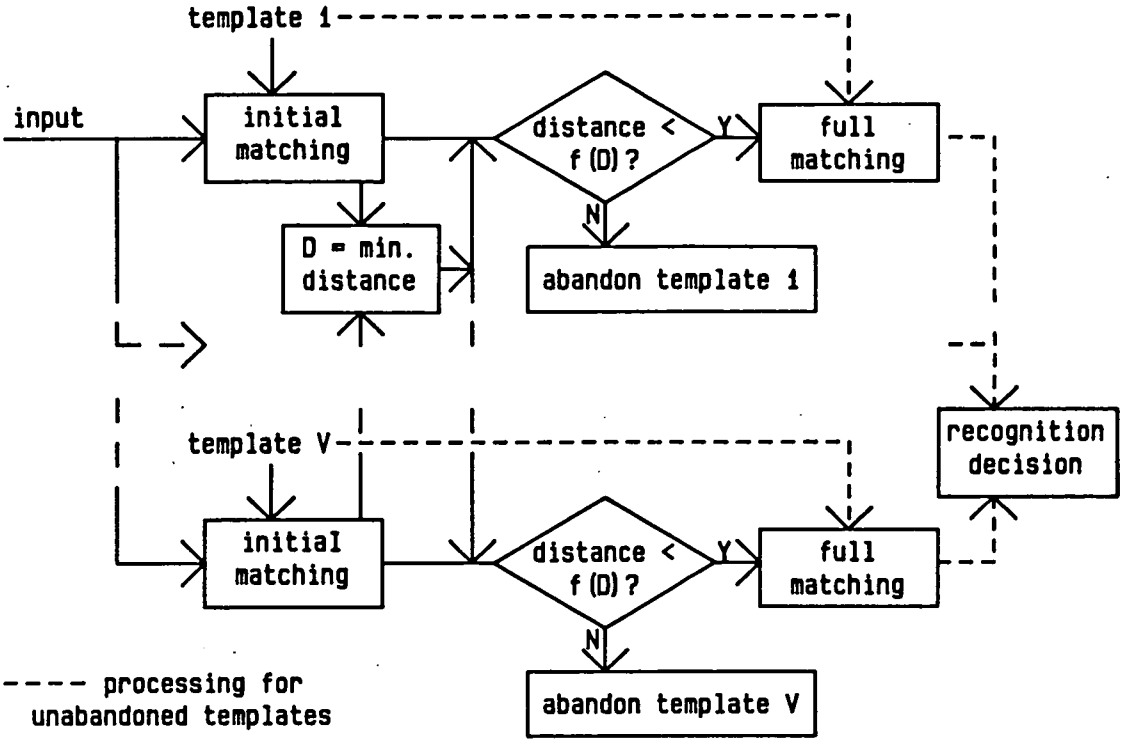
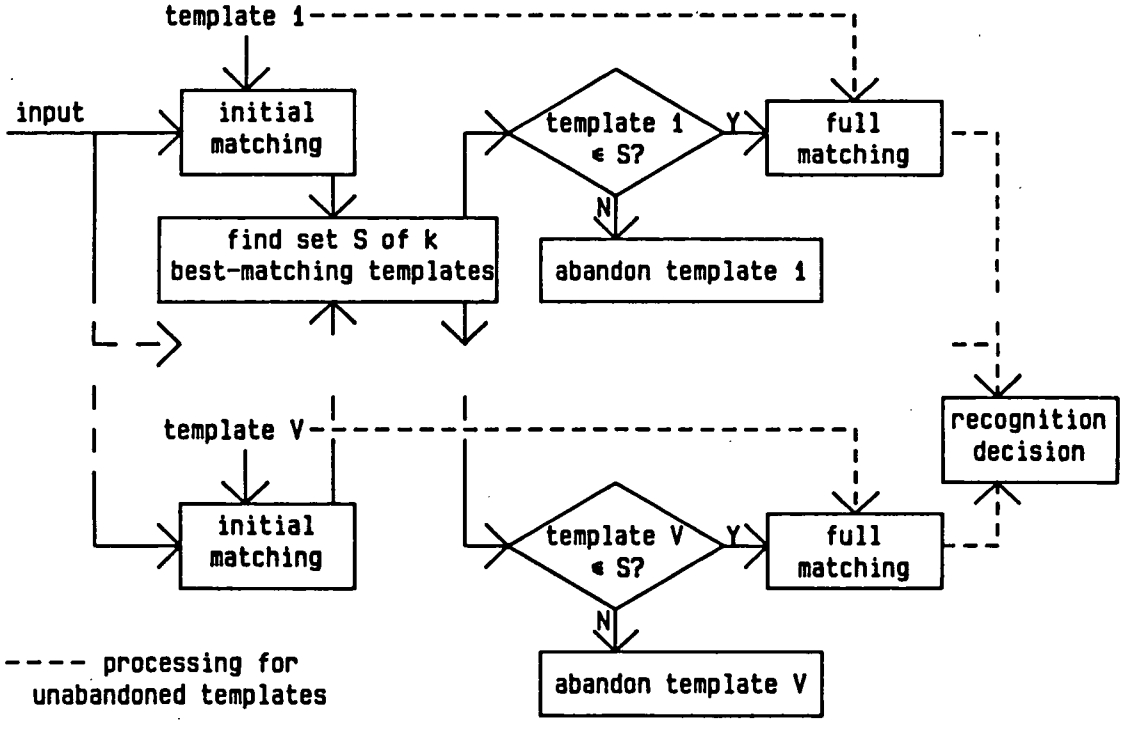


Figure 3.3: elimination of all but a specified number of templates



elimination of all the templates, and hence a "no-recognition" decision, if no template matches the input closely enough – which may or may not be a desirable possibility to allow, depending on details of the system and its application. The second and third types of condition avoid this by making the decision as to elimination of a template depend on its performance relative to the other templates, instead of its absolute performance. They differ in that the retention of a fixed number of templates results in a fixed number of word matching operations to be done, whereas the use of a relative distance threshold allows the number of templates retained, and hence the number of word comparisons, to be increased if there are many closely competing candidates, or reduced if a few templates match the input much better than all the others. If the first or second type of condition is imposed, an absolute or relative distance threshold must be set to give an appropriate combination of computational efficiency and accuracy in recognition.

An isolated word recognition system [255,258,260] which incorporates up to four stages of comparison, with increasing degrees of accuracy and computational expense, is described in chapter 4. This system incorporates trace segmentation or linear time segmentation, as referred to in section 3.2 above, at each stage, and was developed following examination of the results of the comparative experiments with segmentation techniques, described in the earlier part of chapter 4. It uses a condition of the second type above, based on the ratios of the word distances to the best word distance at each comparison stage, to make template elimination decisions. The number of comparison stages, the segmentation used at each stage and the thresholds applied to the word distance ratios can be adjusted to yield any desired tradeoff between speed and accuracy for a given vocabulary and application. Measurements of the recognition times and accuracies attained will be given. On the basis of these results, the

segmentations and threshold values for use in subsequent experiments were chosen, to maintain a level of accuracy similar to that attained without any early elimination of templates, while achieving a substantial reduction in computation time.

3.4: Adaptation of reference patterns during recognition

One respect in which the basic word pattern matching method is clearly deficient, relative to human recognition of speech, is that it derives its information about the words in the vocabulary only from the training utterances, without making any use of the additional repetitions that are made available during a recognition session. Adaptation of the reference patterns (templates or HMMs) to take account of new input data during recognition sessions [238,239,241,246,256] can have several benefits. By adaptation, initial speaker-trained patterns can be made more reliably representative of the speaker's pronunciations, through the incorporation of more data into each pattern; the recogniser can adapt to gradual changes in the speaker's voice or the acoustic background conditions; and, in the case without speaker-specific training, patterns which are initially speaker-independent can be tuned to the characteristics of a particular current input speaker. (It has been observed [228] that a user's voice changes due to fatigue in the course of a long recognition session, or in the course of a day's work during which a speech recogniser is being used intermittently, and this can seriously affect the recognition performance attained using templates formed at or before the beginning of the session. Also, there may be systematic differences between the pronunciations occurring during a training session and those occurring when the system is being used for recognition, because of the differences in the mode of user-system interaction and the

different tasks being undertaken [228,239].)

An adaptive system may modify its existing reference patterns, by an averaging procedure of some sort [238,239,241,256]; it may also include an option for the creation of a new reference pattern where the recognised input is not close enough to any of the existing ones [256].

Adaptation will be most reliable and effective when there is feedback in some form (explicit or implicit) as to the correctness of each recognition. This will allow the appropriate reference pattern to be adapted whenever an input word is correctly identified, while preventing adaptation of a best-matching but incorrect reference pattern. In the absence of such feedback, precautions must be taken [238,239] to prevent the system from becoming unusable because of repeated adaptation of wrongly chosen reference patterns — either by ensuring that such adaptation does not occur or by providing a means to correct the corrupted reference patterns when it does occur. A possible development in the supervised case (i.e. where there is feedback) is adaptation of the best-matching template away from the input word when this word is known to have been recognised incorrectly, with the aim of reducing the likelihood of a recurrence of the same misrecognition. (A training procedure incorporating such adjustment away from incorrectly recognised words has been reported, using weighted averaging with a negative weight [50].) A further development is the verification and adaptation of the template for the second-best recognition candidate, if the first candidate (corresponding to the best-matching template) is incorrect. This may be extended to "full verification" [245], in which the system continues to prompt the user for yes/no responses to all possible recognitions of the input, in order of increasing word distance, until a "yes" response is obtained.

In a multiple-template speaker-independent system, it is possible to apply another form of adaptation [254], in which it is the template set or the decision procedure which is adapted, rather than the individual templates. Templates which are found not to match the current speaker's voice well – such as those from training speakers of the opposite sex – can be eliminated from consideration, reducing the range of possible errors.

In chapter 5, a description is given of template adaptation options which have been incorporated into the multiple-stage isolated word recognition system described in chapter 4 [255,258,259,260]. These permit supervised or unsupervised adaptation, and, in the supervised adaptation case, negative adaptation to misrecognised words and adaptation of the template for the second-best recognition candidate when the best-matching template is incorrect. Two systems of weighting for the weighted averaging of templates and input words are introduced – one which results in tracking of gradual changes in the speaker's voice or the acoustic conditions, and one which results in optimisation of the templates for the speaker's average pronunciations if these are assumed not to vary in time. Various options for control of the adaptation are provided. Chapters 6 and 7 describe experiments with adaptation of speaker-specific and speaker-independent initial templates, and report the effects of this adaptation on recognition accuracy and on the computation required to obtain each recognition. Possible extensions of the adaptation technique are discussed in chapter 8.

3.5: User-system interaction and interface design

To achieve optimal performance of speech recognition systems in practical applications, attention must be devoted not only to the acoustic representation of the speech signal and to pattern-matching techniques, but also to the design of

the whole system into which these will be integrated. The modes of interaction between the system and the user must be so designed as to facilitate the efficient accomplishment of the task or tasks for which the system is to be used.

In section 3.4 above, the possibility of adaptation by the system to the user was introduced. But in any practical speech recognition task there is likely also to be adaptation by the user to the system [228]. This may be beneficial (for example, if the user learns to speak more consistently so as to obtain good recognition performance) or harmful (as in the case where the user becomes irritated by the system's repeated failure to recognise a particular word and begins to shout, making correct recognition of the word even more unlikely). The user-system interface should be designed so as to encourage helpful adaptation and to avoid producing harmful adaptation.

Possible features of such an interface include messages from the system (audible or visual) which guide the user as to mode of speaking (suggesting an increase or decrease in volume or speaking rate for instance), and the provision of convenient means for correcting wrong recognitions and for retraining when particular templates are inadequate [225]. A retraining option is particularly desirable if adaptation is incorporated and the stability of the system cannot be guaranteed. A method of correcting the most recent recognition – whether by saying a designated word such as "correction" or by some other means such as pressing a key – can also allow adaptation to be supervised without the need for an explicit yes/no response from the user to each recognition: the system can be designed to adapt the reference pattern for the recognised word only if the user gives no indication that the recognition is incorrect.

The adaptive multiple-stage isolated word recognition system described in chapter 5 has an interactive mode, in which some of the features mentioned above are incorporated [258,260]. Chapter 6 includes observations and statistics

from interactive recognition sessions, using this system, with several speakers. Some aspects of the interface and interaction between system and user are discussed in chapter 8.

CHAPTER 4

SEGMENTATION AND SEGMENT REPRESENTATION TECHNIQUES, AND THEIR APPLICATION IN A MULTIPLE-STAGE DECISION PROCEDURE

4: SEGMENTATION AND SEGMENT REPRESENTATION TECHNIQUES, AND THEIR APPLICATION IN A MULTIPLE-STAGE DECISION PROCEDURE

4.1: Introduction

Various preprocessing techniques for template-based speech recognition have been devised [60,66,69,76,80,81,82,92,147,152,161,168] which involve time segmentation of each utterance and the derivation of representations for the segmented utterance from the original frame representations. Some of these techniques result in normalisation of each word pattern to a fixed number of vectors [60,76,80,81,82,92,161], while in other cases the number of vectors derived for a word depends on the duration or acoustic characteristics of the word [66,69,80,81,82,147,152,161,168]. The segmentation can be linear [60,81,92,161] (resulting in equal time segments), or defined by some data-dependent rule so that the durations of the segments vary with variations in the rate of change in the characteristics of the speech signal [66,69,76,80,81,82,92,147,152,161,168].

One class of segmentation techniques consists of the various forms of trace segmentation [76,80,81,92,147,152,161] (as described in section 2.3.7 above). The parameters defining a form of trace segmentation include the acoustic representation and distance function used to define and measure the trace in acoustic vector space; the choice of fixed segment length (yielding a variable number of segments per word – hence the term "variable length trace segmentation") or of a fixed number of segments per word ("fixed length trace segmentation"); and the method chosen to derive representations for segments or segment boundaries from the original frame vectors. Only the second of these – the choice of fixed or variable length trace segmentation – has been systematically

investigated in the literature referred to in chapter 2; one of the published comparisons (on several vocabularies) [81] shows some advantage in having a fixed number of segments per word (especially when a strict global path constraint is applied), though other results [80] show a slight disadvantage in this. In experiments with a variable frame rate coding technique which is similar to trace segmentation [82], a fixed number of segments per word was found to be better. The other two considerations – acoustic representation and distance measure, and segment representation – are discussed below.

Some of the previously reported results with trace segmentation [76] were obtained using a spectral representation (derived from a filter bank, without any logarithmic transformation) allowing the use of zero frames at the beginning and end of an utterance to represent silence – which could help to counteract the effects of unreliable endpoint detection whereby the beginnings and ends of utterances might be truncated. Other systems with trace segmentation [80,81,92,147,161] have used log filter energies. One set of results, for connected word recognition, was obtained using a cepstral representation [152]. The distance measure used has been the absolute value metric [76] or the Euclidean metric [80]. (In some cases, the published accounts of the experiments do not state whether the same metric was used to measure the trace as to compute distances in the subsequent matching; but the distance function used for the latter purpose was the absolute value metric [81,147,161] or in one case the squared Euclidean metric [152].)

Once the trace has been defined (by the choice of acoustic analysis and distance measure) and segmented, it remains to derive a sequence of acoustic vectors from the segmented utterance. This may be done by some technique which yields a vector for each segment, or by one which produces vectors corresponding to the segment boundaries. Three different segment representation techniques

have been used by previous experimenters. The first [76,80,161] consists of linear *interpolation* of a vector at each segment boundary. This yields $S+1$ vectors, where the number of segments is S – provided that the initial and final vectors, corresponding to the start and end of the trace, are included. The second technique [81,82,147] is *selection* of the nearest of the original frame vectors to each segment boundary. This yields an approximation to the representation that would be obtained by interpolation; again the result is $S+1$ vectors, one for each segment boundary. The third technique [152] derives a vector for each segment (rather than for each segment boundary), by *averaging* of all those original vectors which occur in that segment of the trace. In this case the number of vectors resulting is S. If it is possible for a segment to occur which does not include any of the original vectors (because it lies entirely between two successive frame vectors on the trace in the acoustic parameter space), then some procedure to cope with this will be required – such as an adjustment of the segmentation to ensure that every segment contains at least one original vector, or the derivation of a vector by some other method where the usual averaging is not possible.

Most of the comparisons of trace segmentation with linear time segmentation show that the acoustically determined non-linearity introduced by the trace segmentation yields an improvement in recognition performance over that obtained with linear segmentation [81,92,161]. The exception is in Ney's experiments with connected German digits, where the error rates were similar with trace segmentation and with linear downsampling [152]. There are several possible reasons for this apparent inconsistency: it could be a result of the choice of acoustic representation (cepstral coefficients rather than log filter energies), the distance function used to measure the trace, the segment representation technique (averaging instead of interpolation or selection), or some effect of the

vocabulary and speakers.

In the next section of this chapter (section 4.2), some experiments with trace segmentation and linear segmentation (with a mel cepstral representation, a fixed number of segments per word and the three different segment representation techniques) are described, and the results are presented and discussed. Section 4.3 describes an isolated word recognition system with a multiple-stage decision procedure, which combines the computational economy of template matching with a small number of segments per word and the recognition accuracy possible with larger numbers of segments; and section 4.4 gives the results of tests of the performance of this system to determine appropriate values of the system parameters. Section 4.5 summarises the results in the chapter.

4.2: Segmentation and segment representation experiments

The results reported in the sections below have previously appeared in a paper [97] which is reproduced at the end of this thesis. Minor errors in the plots of results in that paper have been corrected in figures 4.1 to 4.4 below.

4.2.1: Speech data base

Two vocabularies were used in these experiments: the digits (0 to 9, with 0 pronounced "zero"), and a vocabulary of 20 mostly polysyllabic words (listed in table 4.1). (The words in the second vocabulary were chosen from among those occurring in the "golden passage" selected for use in the Edinburgh University Speech Input Project, and will be referred to below as "the GP vocabulary".)

Three speakers – two male (speakers 1 and 2) and one female (speaker 3) – provided utterances of the words of these two vocabularies. The training data for each speaker consisted of five repetitions of the digits (in order from 0 to 9)

Table 4.1: words in GP vocabulary

against	framework	retaining	these
begin	horizontal	single	those
evergreen	Japanese	sometimes	trained
flowering	possible	spring	training
following	remaining	susceptible	year

and three repetitions of the GP vocabulary (in alphabetical order). Each speaker also spoke each vocabulary five times (in five different random orders) to provide test data. The utterances were collected in a recording studio using a high quality boom-mounted microphone, and recorded digitally, and were subsequently transferred to analogue tape, lowpass filtered at 5kHz and digitised at a 10kHz sampling rate. The detection of word endpoints was accomplished by visual inspection of waveform displays.

The endpoint-detected words were analysed to obtain eight cepstral coefficients, based on a simulated filterbank with bandpass filters on a mel frequency scale [23], for each frame of speech, where the interval between successive frames was 12.8ms. The number of frames per word ranged from 16 to 57 (with average 33.8) in the case of the digits, and from 23 to 75 (average 45.5) for the GP words. The range of durational variation for utterances of any one word was considerably narrower, however. For the three speakers, the average lengths of the digits were 30.8, 32.2 and 38.4 frames, and the average word lengths for the GP vocabulary were 40.4, 46.7 and 49.5 frames, respectively.

4.2.2: Segmentation and recognition of words

After the processing described above, each word was represented by a sequence of vectors of eight floating point numbers, each vector consisting of the first eight mel cepstral coefficients derived from one frame of speech. For recognition without segmentation, these sequences of vectors were used directly in the

DTW matching process. For the experiments with segmentation, they were taken as the input to the segmentation procedure, which generated a sequence of vectors for each word, to be used in the DTW matching.

A Fortran program was composed to perform segmentation (optionally) and recognition. Options available in the segmentation stage included the choice of fixed segment length or of a fixed number of segments per word; the absolute value or Euclidean metric for measuring the trace; and the interpolation, selection or averaging form of segment representation. There was also a linear time segmentation option, with a choice of fixed segment duration or of a fixed number of segments per word, again with the three possible forms of segment representation.

The recognition stage used Itakura's form of local path constraints and weighting scheme (c) (as described in sections 2.3.3 and 2.3.4), with the input word on the horizontal axis. The frame distance function used was the absolute value metric. The global constraints could take the form of a parallelogram (with sides of slope 0.5 and 2.0 – corresponding to the minimum and maximum gradients allowed by the local path constraints), a band of some specified width centred on the line of slope 1.0, or a band centred on the linear path from the initial to the final point (as described in section 2.3.6); the second and third of these options are identical if the input word and template are normalised to the same length. No endpoint relaxation was employed.

In the averaging form of segment representation, when a segment of the trace occurred which did not contain any of the original frame vectors, the segment was extended as far as the next frame vector, which was then used as the representation for that segment; each subsequent segment was then defined, as usual, by measuring the previously determined distance ($\frac{T}{S}$ in the notation of

section 2.3.7) along the trace from the end of the previous segment. Thus it was possible, with the averaging option, for the number of segments, and hence of vectors after segmentation, to be less than the specified number S. This was particularly liable to occur where S approached or exceeded the number of frames in the word.

4.2.3: Recognition performance measures

In the experimental evaluation of segmentation and segment representation techniques as preprocessing for DTW-based word recognition, three measures of word recognition performance were used.

The first measure was the rate of correct recognition, expressed as a percentage over all the trials for a given set of segmentation conditions.

Secondly, a "recognition quality measure", R, was defined, as follows. For one recognition of one utterance, let r be the ratio of the smallest word distance obtained for an incorrect template to the word distance obtained for the correct template. (If $r>1$, the correct template is the nearest template to the test word, and so the word is recognised correctly. If $r<1$, the best-matching incorrect template is closer than the correct one, and so the word is misrecognised.) Then, for a given set of word recognitions,

$$R = \frac{(\text{mean value of } r) - 1}{\text{standard deviation of values of } r}. \qquad (4.1)$$

R is a measure of the ability of the recognition system, with the segmentation used, to discriminate correct and incorrect templates. If the distribution of values of r is assumed to be of a similar shape for each set of segmentation conditions, and to differ only in its mean and standard deviation, then the value of R is monotonically related to the expected recognition accuracy, which is the

probability that $r > 1$. The advantage of R over the simple recognition rate as a measure of performance is that it makes fuller use of the word distance information derived from a set of experiments, and so can provide a reliable measurement from a relatively small number of recognition trials. Measuring performance by the recognition rate is equivalent to quantising r to only two values, corresponding to cases where $r > 1$ (correct recognition) and where $r < 1$ (incorrect recognition). The disadvantages of using R values alone to measure performance are that R does not give an explicit estimate of recognition accuracy (since the relation between R and the recognition accuracy depends on the shape of the distribution of r), and that the monotonic relation between R and expected accuracy breaks down if the shape of the distribution of r is not constant across different segmentation conditions. It should be noted, also, that the relation between R and the expected recognition accuracy is not linear (the recognition accuracy is a concave function of R, assuming a consistently-shaped unimodal distribution for r, as long as the modal value of r is greater than 1), and so the recognition accuracy corresponding to the average value of R (taken from the individual R values for a population of template sets, or of speakers) will not usually be equal to the average recognition performance for the same population: averaging values of R and reading off the corresponding expected recognition accuracy will tend to yield an optimistic estimate.

The third measure adopted was based on the idea of setting a threshold on the ratio of the word distances for the best two templates (i.e. those closest to the current test word), so as to give a "rejection" response instead of a recognition in cases where the decision between these two recognition candidates was uncertain (as indicated by a ratio too close to 1.0). The performance measure calculated was the rejection rate corresponding to the minimum threshold value required to reduce the rate of wrong recognitions to 2%. This measure has the

disadvantage that it is sensitive to the value of one particular word distance ratio in a set of recognitions, namely the ratio (for an incorrect recognition) which determines the setting of the threshold. However, it complements the recognition-rate measure by giving an indication of how well the recogniser would perform with a rejection option implemented. A more sophisticated rejection-rate measure could be constructed by accumulating the rejection rates corresponding to several different rates of incorrect recognition, instead of using only that corresponding to 2%; however, this development was not pursued in these experiments.

4.2.4: Preliminary experiments and setting of fixed parameters

Informal experiments were carried out on subsets of the data base to determine the effects of different settings of some of the options described in section 4.2.2. Appropriate settings of these options were thus determined, and these were used throughout the main series of experiments.

No advantage was detected in using the Euclidean metric, rather than the simpler absolute value metric, to measure the trace. Accordingly, the absolute value metric was chosen for subsequent use. Also, an analysis of the distances obtained for correct templates and for best-scoring incorrect templates revealed no advantage in using the squared Euclidean metric instead of the absolute value metric as the frame distance function in the DTW matching procedure, and so, here also, the absolute value metric was adopted.

It was found that no major reduction in computation could be achieved by applying global path constraints, without diminishing the recognition accuracy from that attained with the full parallelogram in the input-reference plane. The parallelogram constraints were therefore adopted.

One other parameter choice adopted for the main series of experiments was the specification of a fixed number of segments per word, rather than a fixed segment length.

4.2.5: Experiments

In the main series of experiments with segmentation and segment representation techniques, recognition performance was evaluated on the whole data base (with the two vocabularies treated separately) for each set of segmentation conditions (with certain exceptions as noted below). The segmentation conditions were defined by the choice of trace segmentation or linear time segmentation; the number of segments per word (S); and the choice of segment representation technique (interpolation, selection or averaging). The performance with no segmentation was also evaluated. All the experiments used speaker-specific templates: no cross-speaker recognition results were obtained.

For each set of segmentation conditions, and for each speaker, recognition experiments were carried out as follows.

In the case of the digits, each of the five training repetitions of the vocabulary in turn was used as a template set for recognition of the five test repetitions. The value of R was computed for each set of 50 recognitions with one template set, and the values of R and rates of correct recognition for the five template sets were averaged. The rejection threshold for a 2% misrecognition rate was determined by examination of the full set of 250 ratios of second-best to best template distances (pooled across the five template sets), and the rejection rate was computed using this threshold value.

In the case of the GP vocabulary, a similar procedure was followed, but in this case there were only three sets of templates (from the three training

repetitions) per speaker, and the test data consisted of 100 words (five repetitions of the 20 words in the vocabulary). Thus each R value was computed from 100 recognitions, and the rejection threshold and rate were determined from 300 recognitions.

In the experiments with the digits, the number of segments per word, S, was varied from 1 to 30 in the case with averaging, and from 6 to 31 in the cases with interpolation and selection. For the GP words, S was varied from 1 to 45 (linear segmentation) or 46 (trace segmentation) with averaging, and from 9 to 45 with interpolation or selection. (It was found that the performance with interpolation or selection was much poorer than with averaging when S was small, and so full experiments with small values of S were performed only with averaging.)

Because the recognition performance with selection was found to be consistently poorer than with interpolation, when evaluated on the digits and on one speaker's utterances of the GP vocabulary, the selection technique was not applied to the utterances of the GP words by the remaining two speakers.

4.2.6: Results

The results obtained with the various segmentation and segment representation techniques are presented in figures 4.1-4.4.

Figures 4.1 and 4.2 show the results for recognition of digits using linear time segmentation and trace segmentation respectively. For each set of segmentation conditions, the recognition error rate and the value of the recognition quality measure R (each computed as described above and then averaged over the three speakers) are plotted against the average number of vectors per word. (The rejection rate for 2% error was found to be subject to a greater degree of

Figure 4.1: recognition results for digits using linear time segmentation

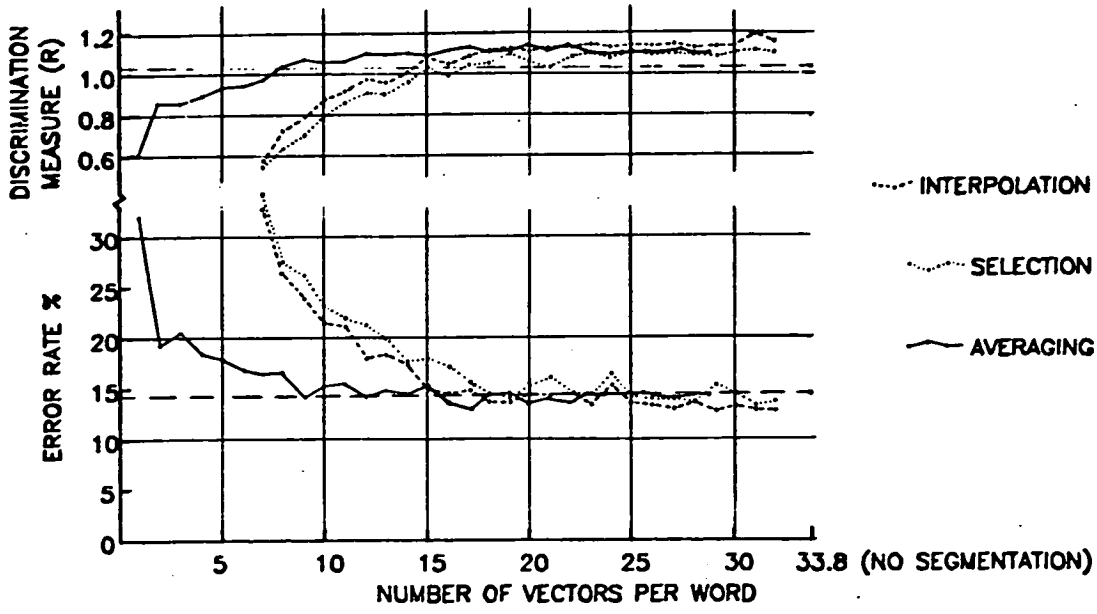
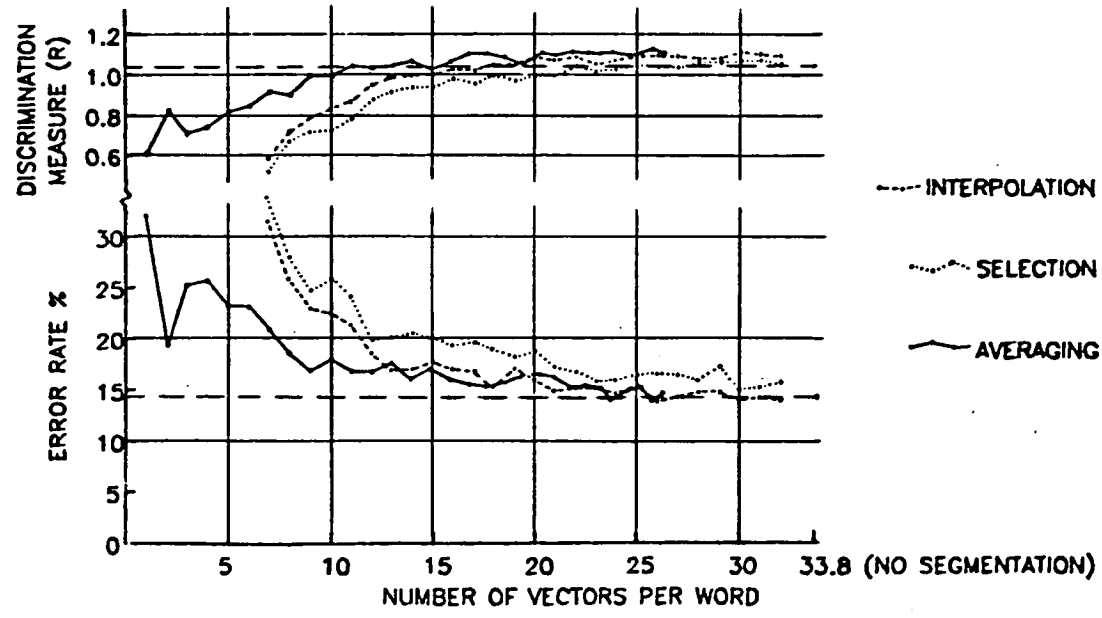


Figure 4.2: recognition results for digits using trace segmentation



irregular variation than either of the other two measures, and is therefore not plotted.) The results with averaging, interpolation and selection are joined by solid, broken and dotted lines respectively. The results without segmentation are plotted (against the average number of frames per word) at the right of each figure, and are also marked by horizontal lines to facilitate comparison of performance with and without segmentation.

It will be noticed that the points corresponding to the averaging technique for segment representation are not all at integer numbers of vectors per word. This is because the segmentation adjustment adopted (as described in section 4.2.2) allows the number of segments in a word to be reduced in cases where some segment contains no frame vector. With this adjustment, the number of vectors per word after segmentation need not be the same for all words, for a given value of S, but will vary according to the length of each word and (in the case of trace segmentation) according to the positions of the frame vectors along the trace. Thus the average number of vectors per word is liable to be less than S, and will not in general be an integer, especially when S approaches or exceeds the number of frames in the shortest word.

Figures 4.3 and 4.4 show the corresponding results for the GP vocabulary, with the averaging and interpolation techniques only.

4.2.7: Discussion of results

4.2.7.1: General comments

The overall average level of accuracy attained was rather poor, especially for the digits, where the average recognition rate without segmentation was only 85.6%. (The corresponding recognition rate for the GP vocabulary was 87.2%; but the performance on this vocabulary might have been expected to be poorer

Figure 4.3: recognition results for GP vocabulary using linear time segmentation

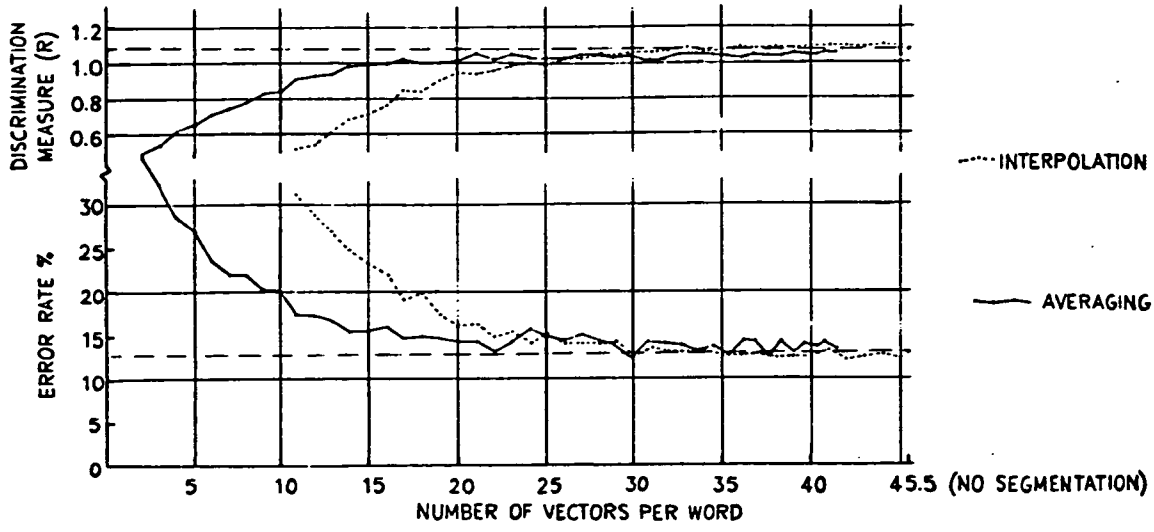
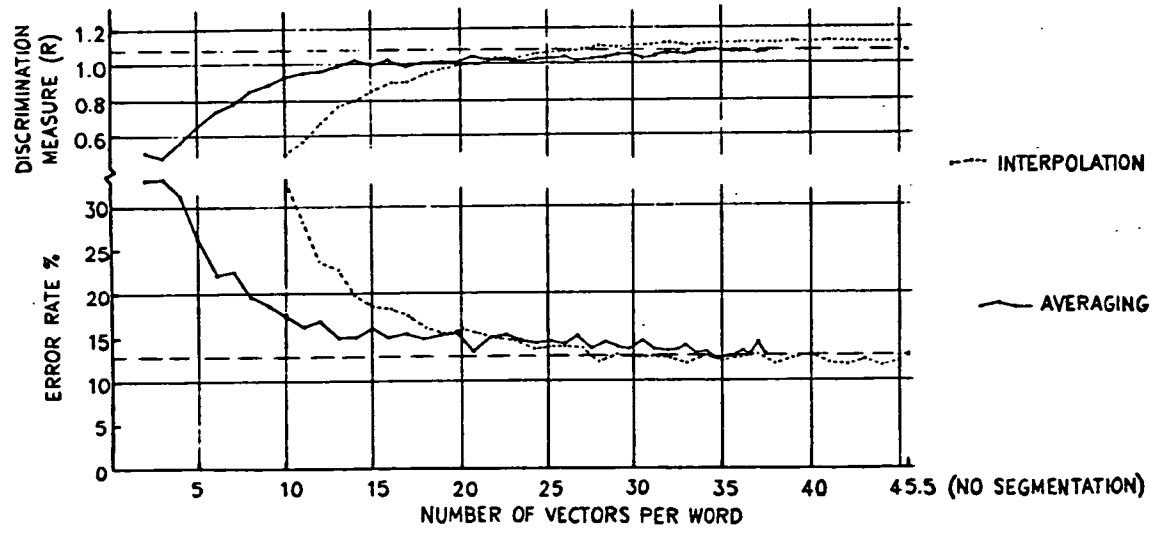


Figure 4.4: recognition results for GP vocabulary using trace segmentation



because of the larger vocabulary size and the occurrence of confusable word pairs such as {remaining,retaining} and {flowering,following}, whereas the digits are generally found to be a fairly easy vocabulary for recognition.)

A more detailed examination of the results for individual speakers and sets of templates reveals considerable variations from one speaker to another, and also, in some cases, from one template set to another for a given speaker. The average recognition rates for the three speakers were 93.2%, 75.2% and 88.4% for the digits, and 85.3%, 87.7% and 88.7% respectively for the GP vocabulary. The recognition accuracies for the individual template sets are shown in table 4.2. (All these results are for the case without segmentation. The variability was similar in cases with segmentation.)

The poor results obtained using some sets of templates, such as the first four sets of digit templates for speaker 2, suggest that a major reason for the overall low recognition accuracy is that certain templates did not well represent the pronunciations of the words occurring in the test input – perhaps because of inconsistent pronunciations by the speakers, inclusion of non-speech sounds, or inaccurate placing of word endpoints. (The endpoint location was done entirely on the basis of visual inspection of waveform displays, without any auditory checking, which could lead to the omission of low-amplitude fricative sounds at

Table 4.2: recognition accuracies for individual template sets

Speaker	Recognition accuracies for template sets (%)							
	Digits					GP words		
1 (m)	92	96	90	96	92	90	79	87
2 (m)	72	78	66	70	90	87	90	86
3 (f)	88	90	88	90	86	82	91	90

the beginnings and ends of words.) This explanation is supported by the observation that the recognition errors occurring for a given template set often consist mainly of repeated occurrences of a small set of misrecognitions (e.g. 5-9, 7-2 and 8-3 in the confusion matrix, shown in table 4.3, for the second digit template set from speaker 2), and the distribution of such word confusions occurring varies among the different template sets for one speaker although the input for recognition is kept the same.

However, the aim of these experiments was to assess the relative performance of different forms of segmentation and segment representation, rather than to attain an optimal level of performance as expressed in absolute recognition scores. The presence of some bad templates introduces statistical "noise" into the results, but does not invalidate the comparison of techniques, especially as the same phenomena of variable pronunciation, non-speech sounds and inaccurate endpoint detection are liable to occur in practical applications.

Table 4.3: confusion matrix for a template set including some bad templates (speaker 2, digits, template set 2)

	Word recognised									
Word spoken	0	1	2	3	4	5	6	7	8	9
0	5	0	0	0	0	0	0	0	0	0
1	0	5	0	0	0	0	0	0	0	0
2	0	0	5	0	0	0	0	0	0	0
3	0	0	0	5	0	0	0	0	0	0
4	0	0	0	0	5	0	0	0	0	0
5	0	0	0	0	0	3	0	0	0	2
6	0	0	0	0	0	0	5	0	0	0
7	0	0	4	0	0	0	0	1	0	0
8	0	0	0	5	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	5

Another general observation on the results is that the performance obtained with segmentation, with interpolation and a number of segments per word which approaches the average word duration in frames, is usually better than the performance with no segmentation. This can be seen most clearly from the plots of R in figures 4.1-4.4. Two possible factors contributing to this are the beneficial effect of word length normalisation (observed by some previous experimenters [60,81], though not by others [80]) and the smoothing effect which occurs with interpolation (as considered in section 4.2.7.3 below).

As the number of vectors per word, obtained by segmentation, is reduced, the recognition accuracy attainable (with appropriate choices of segmentation and segment representation techniques) declines only slowly, until the number of vectors reaches about one third of the average number of frames per word. This is true particularly in the case of digit recognition, where the temporal structures of the words are simpler than in the other vocabulary: accuracies similar to that without segmentation can be obtained with as few as 9 vectors per word, as seen in figure 4.1. Even with only 2 vectors per word, derived by averaging, accuracies of 80.8% (digits, linear time segmentation) and 67.0% (GP words, trace segmentation) were obtained.

4.2.7.2: Comparison of trace segmentation and linear time segmentation

In general, little difference was observed between the performance figures for trace segmentation and for linear time segmentation. This is in agreement with the results of Ney [152], though not with those of other researchers [81,92,161] who were using acoustic representations other than cepstral coefficients.

There was, however, a tendency for the results on the digits to be better with linear time segmentation, and for the results on the GP words to be better with trace segmentation. These differences between trace segmentation and linear time segmentation results were not observed uniformly across all segment representation techniques and numbers of segments per word.

In the case of the digits, the difference was largest and most statistically significant with the averaging method of segment representation and a small number (up to about 10) of segments per word: the average recognition accuracy for 4 to 7 segments per word was 5.87% higher with linear time segmentation than with trace segmentation, and the standard error of this figure, estimated from the variation across speakers, was 1.52 — which yields the conclusion that linear time segmentation is better than trace segmentation, for this vocabulary and these numbers of segments, with confidence 0.97. (Details of the statistical analysis applied — which incorporates a one-tailed t test — can be found in the appendix.) The corresponding average difference and standard error estimate for 8 to 12 segments per word were 2.19% and 1.09 respectively, corresponding to confidence 0.91. Over all numbers of segments per word, the average difference was 2.05% (standard error 1.33, confidence 0.87) with averaging, and 0.85% (1.09, 0.74) with interpolation.

In the case of the GP vocabulary, the difference between trace segmentation and linear time segmentation was less consistent than for the digits, but a statistically significant difference was found with small numbers (11-15) of segments per word and the interpolation technique: here the accuracy with trace segmentation was 4.56% higher, and the standard error estimate for this difference was 0.33, yielding a confidence level of 0.997. The difference between trace segmentation and linear time segmentation results over larger numbers of segments per word (24-43) was less significant: over this range the average

difference, again with interpolation, was 0.60%, with standard error estimated at 0.69, and hence confidence 0.76. The differences, standard error estimates and confidences over all numbers of segments were 1.36% (0.63, 0.92) with interpolation and 0.20% (0.34, 0.69) with averaging. With averaging, as these figures indicate, the difference was less consistent: in some cases the results with linear time segmentation were better than those with trace segmentation.

A possible reason why trace segmentation is better (relative to linear time segmentation) for recognition of the GP vocabulary than for digit recognition is that this vocabulary contains more disyllabic and polysyllabic words, in which the range of non-linear timescale variability is likely to be greater than for monosyllabic words, and so the non-linear adjustment of the timescale (in accordance with the acoustic structure of the word) provided by trace segmentation will tend to be more helpful. This does not, however, account for the fact that the performance on the digits is actually poorer with trace segmentation than with linear time segmentation.

A possible explanation for the results with the digits is that trace segmentation tends to emphasise the transitional or varying portions of speech rather than the steady portions: if, because of features of the vocabulary or of the speakers, the steady portions, such as vowels, are the main recognition cues, or if non-speech sounds (such as breath noise) are included and these have a high degree of frame-to-frame variability, the emphasis on the more rapidly varying portions may worsen the recognition. If this phenomenon were due to speaker characteristics or non-speech sounds, it should occur both with the digits and with the GP words, but the effect might be cancelled out in the case of the latter by the timescale variability compensation referred to above.

These explanations of the results remain conjectural, in view of the degree of variability occurring among speakers and among segment numbers and segment representation techniques, and the small number of speakers used in the experiments. However, it does appear that trace segmentation is not greatly superior to linear time segmentation when applied to a mel cepstral representation of speech, at least for recognition of the digits vocabulary; and that its usefulness depends on vocabulary characteristics.

4.2.7.3: Comparison of segment representation techniques

The comparative results obtained for the three segment representation techniques are more consistent, and easier to account for, than those for the two segmentation techniques examined above.

The averaging of the vectors within each segment gives better recognition results than either interpolation or selection of vectors at segment boundaries when the number of segments per word is small. This is what might be expected, since averaging makes full use of all the original vectors, whereas interpolation uses only the two neighbouring vectors at each segment boundary, and selection uses only one of these. With the interpolation or selection technique, when a segment contains three or more vectors – which occurs frequently when the number of segments per word is small – the intermediate vectors in the segment are ignored once the trace (or for linear time segmentation the timescale) has been defined and measured.

When the number of segments per word approaches the number of frames, interpolation becomes better than averaging. This may be partly because interpolation results in a more precise representation of the progress along the trace (in trace segmentation) or of the timescale: the interpolation operation is a

weighted averaging of two vectors, in which the weights depend on their relative distances from the segment boundary, whereas in the averaging technique all the vectors in a segment are assigned equal weight regardless of their precise positions relative to the segment's boundaries. But also, when the number of segments exceeds half the number of frames, interpolation results in more effective smoothing of the original vector sequence than averaging: each interpolated vector is the weighted average of two original vectors, whereas, because there are some segments which contain only one vector, with the averaging technique there will be points at which no smoothing of neighbouring vectors occurs.

A feature of the averaging technique used in these experiments is the adjustment of the segment boundaries when a segment contains none of the original vectors. This may not be the optimal way to handle such cases: some other technique, such as interpolating a vector at the centre of the segment, might be better. This too affects mainly the results with large numbers of segments per word.

The performance with selection is consistently slightly poorer than with interpolation. (The mean differences, over all number of segments per word, for the digits are 1.21%, with linear time segmentation, and 2.02%, with trace segmentation; the corresponding estimated standard errors are 0.44 and 0.93 respectively, giving confidences 0.94 and 0.92. For the GP words, the mean differences for the one speaker on whose data the selection technique was evaluated are 2.51% for linear time segmentation and 3.25% for trace segmentation.) This also is what might be expected: the position, on the trace or on the timescale, of the nearest original vector, adopted in the selection technique, is only a rough approximation to the position of the segment boundary at which a vector would be interpolated; and the smoothing effect of the weighted averaging used in interpolation is not obtained by selection, leaving a more sensitive

dependence on individual vectors in the original sequence, which are subject to unpredictable variations.

4.3: Design of a multiple-stage decision procedure

4.3.1: General features of the design

The observations as to the performance attainable with a small number of vectors per word (as reported at the end of section 4.2.7.1 above) led to the idea of using segmentation in a multiple-stage system, with an initial coarse and computationally simple comparison using a small number of segments per word, to eliminate the most unlikely candidate templates, followed by successively more detailed comparisons using larger number of segments per word to make the choice among the remaining candidates. It was envisaged [97] that a slightly higher recognition accuracy might be attained by such a system than by a single-stage recognition system using DTW without segmentation, because the final stage of the multiple-stage system could incorporate segmentation with a large number of segments per word, which had been observed to provide an improvement in performance over the no-segmentation condition.

Hierarchical decision procedures incorporating similar strategies had been implemented previously by various researchers, as described in section 3.3. However, the system described here based on segmentation has certain advantages of flexibility and extensibility over some of the hierarchical systems developed elsewhere.

Most hierarchical decision procedures for isolated word recognition have just two stages – a pre-matching stage to eliminate unlikely recognition candidates, and a main comparison stage to make the decision among the remaining candidates. The system described here, however, can operate with any number

of stages, allowing as many levels of pattern discrimination as may be required between the coarsest (and quickest) and the finest (and most time-consuming). In the present implementation, up to four stages can be accommodated, but this maximum could easily be increased.

Each stage of the decision procedure uses the same basic techniques – segmentation of the input and templates, and DTW matching. It is only the parameters such as the number of segments per word and the choice of segment representation technique which change from one stage to the next. Thus the same software modules can be applied at all the stages, with only changes in the arguments that are supplied to them. (In a hardware implementation, similarly, the same specialised processor or processors might be used at all the stages, provided that the processor architecture allowed rapid matching of many successive templates when the number of vectors per word was small.)

Because the same modules are used at all stages, the system configuration is easily programmable: both the number of stages and the parameter values within each stage can be redefined each time the system is used, to give an appropriate tradeoff between speed and accuracy of recognition for any vocabulary and application. The parameters at each stage include not only the segmentation conditions and details of the DTW algorithm, but also the thresholds applied to the ratios of word distances for rejection of the input and for elimination of poorly-matching templates.

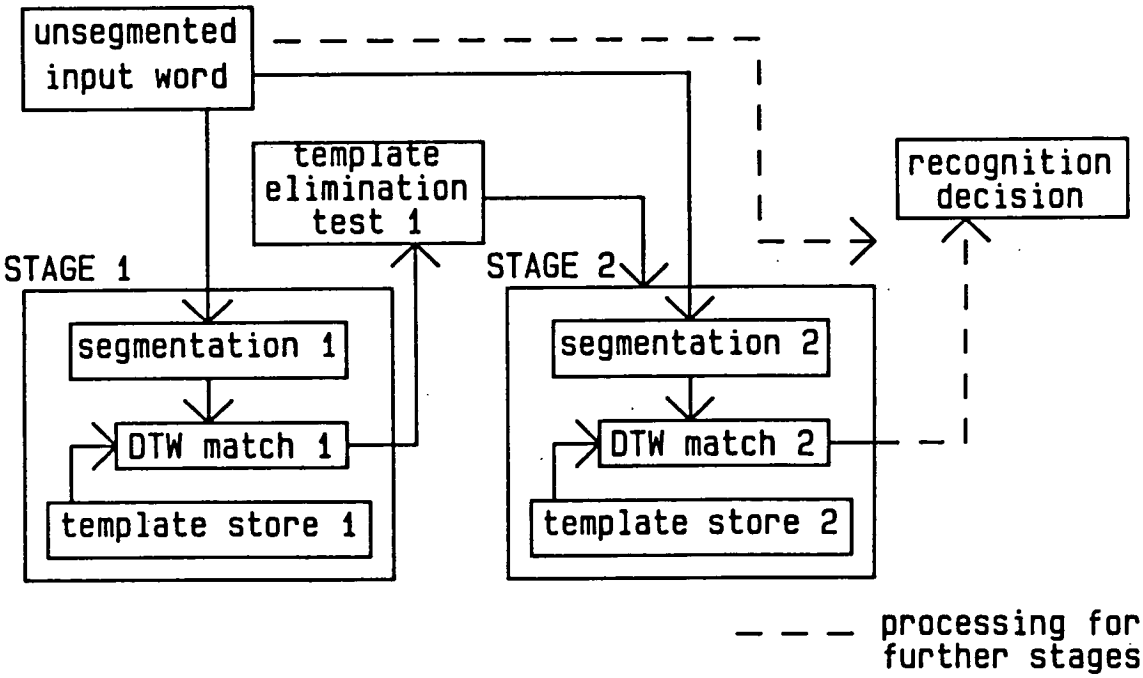
As mentioned in section 3.3, the decision rule for elimination of templates at each stage is based on the ratios of the distances obtained to the distance for the best-matching template. This allows the number of templates retained at the next stage to vary according to the number of plausible candidates. Thus the same decision threshold settings can be retained for vocabularies of varying size and difficulty, without resulting in unnecessary amounts of computation for

small and easy vocabularies, or in loss of accuracy on larger and more difficult ones. Also, for any given vocabulary, the system will adjust the amount of processing appropriately for recognition of the more confusable and less confusable words within the vocabulary. As many or as few templates will be retained at the later stages as the difficulty of each recognition requires.

4.3.2: System and implementation details

The structure of the multiple-stage recognition system (programmed in C on a Masscomp MC550 computer) is shown in figure 4.5. The number of stages is programmable and can range from 1 to 4. The structure of each stage is the same, as shown for the first and second stages.

Figure 4.5: multiple-stage recognition system

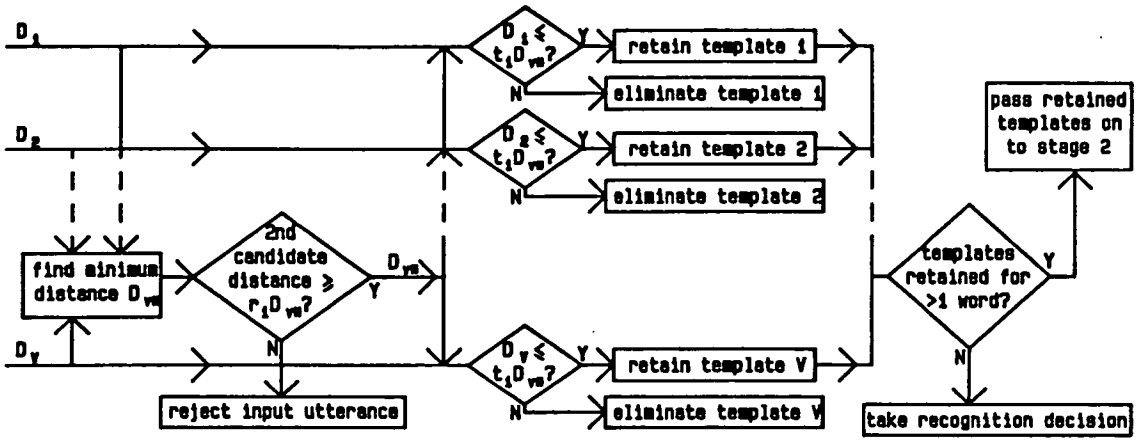


At the start of a series of recognition trials, the previously stored templates are loaded in, and segmented versions of them are derived for use in each word comparison stage. Normally the segmentation at the first stage uses a small value of S so as to generate a small number of vectors per word; the compressed versions of the templates resulting from this segmentation are kept for use in the coarse initial (first-stage) comparison. The segmentation at the second stage has a larger value of S, to produce less severely compressed forms of the templates which will facilitate a finer and more computationally complex comparison. The segmentations at later stages, if these are in use, have successively larger S values. It is also possible for one stage to have no segmentation, in which case the templates are kept in their original forms for use at that stage.

Once the templates have been segmented, the input patterns, representing the utterances to be recognised, are processed sequentially. For each input utterance, the processing is as follows.

The first-stage segmentation is applied to the input to generate a reduced version for use in the first-stage comparison. (The original unsegmented input pattern is kept to be used in deriving further versions for matching at the later stages if required.) This segmented version of the input is then compared by the DTW algorithm with each of the similarly segmented templates already derived. This comparison results in a set of word distances, one for each template. These distances are used in the first-stage template elimination procedure, which has the structure shown in figure 4.6. The minimal distance D_{v*} is identified, where D_v is the distance for template v and v^* is the index of the template best matching the input at this stage. A threshold t_1 is set on the ratios of the distances to D_{v*}: if

Figure 4.6: template elimination procedure after the first stage of comparison



$$\frac{D_v}{D_{v^*}} > t_1, \tag{4.2}$$

template v is eliminated, and otherwise it is retained. A threshold r_1 (which should be lower than t_1) may also be imposed, to allow rejection of the input utterance if the second-best recognition candidate yields a distance too similar to the minimal distance D_{v^*}: the input is rejected if

$$\frac{D_v}{D_{v^*}} < r_1 \tag{4.3}$$

for some template index v such that template v represents a different word of the vocabulary from template v^*. (Some care must be taken in stating this criterion: if several templates are in use for each word of the vocabulary, there will be template indices v other than v^* which correspond to templates for the same word, and the distances D_v for such templates can be close (or even identical) to

D_{v^*} without making the identification of the word unreliable.) Normally, however, r_1 is set to 1.0, so that a rejection decision cannot be taken at the first stage.

If all but one (the one corresponding to the template index v^*) of the words in the vocabulary have been eliminated from consideration by the first-stage template elimination procedure, the recognition decision is taken at this stage, without recourse to the more detailed comparison available in the second and subsequent stages. Otherwise (and assuming that a rejection decision has not been reached), the second-stage versions of all the templates not eliminated are used in the second stage of word comparison.

The second stage is similar to the first: a second segmented version of the input pattern is derived, and this is compared by DTW with the second-stage version of each of the templates not eliminated from consideration at the first stage. If only two stages are in use, the recognition (or rejection) decision is taken on the basis of the distances obtained at the second stage. Otherwise, a template elimination procedure, similar to that at the first stage, but with threshold values t_2 and r_2, is applied, using the second-stage distances. Again, a recognition decision, or possibly a rejection decision, may be reached, in which case the third stage and (if it exists) the fourth stage are not required. Otherwise, the third stage is invoked, to make a more detailed comparison of the input with all the templates not eliminated at the first or second stage.

This process of progressive comparison and template elimination continues until a recognition (or rejection) decision is made, either by elimination of all candidates but one (or fulfilment of the rejection condition) at some non-final stage or else by the nearest neighbour criterion (or, again, fulfilment of the rejection condition) at the final stage. (The rejection criterion at the final stage takes the same form as the criterion for rejection at a non-final stage: this is the

same as (4.3) except for the threshold value r_1 which becomes r_n at stage n.)

The parameters to be defined for each stage of the procedure fall into three groups: segmentation parameters, DTW algorithm settings and decision thresholds. The segmentation parameters are the choice of segmentation technique (trace segmentation, linear time segmentation or none) and (except for the case with no segmentation) the value of S and the segment representation technique. The DTW algorithm parameters include the global constraints (with the option of restricting paths to a band of some specified width in the input-reference plane), the presence or absence of endpoint adjustment (described below), the choice of the absolute value metric or squared Euclidean metric as the frame distance function, and the parameters of an accumulated distance threshold function [12,54] to prevent full matching of templates yielding large distances. Two decision thresholds are defined for each non-final stage: at stage n these are t_n (the template elimination threshold) and r_n (the rejection threshold). At the final stage, only a rejection threshold (r_n) is defined.

The endpoint adjustment technique included as an option is a variation on the technique of Haltsonen [91] using initial and final silence frames. The difference between the silence frame technique and that adopted here is that the latter has, in place of a vector representing silence or background noise, a one-frame pseudotemplate, as used in some connected word recognition algorithms [148,162]. When an input vector is matched to the pseudotemplate, a fixed distance is generated, regardless of the acoustic parameter values composing the input vector. The initial and final pseudotemplates are intended to match any intervals at the beginning and end of the input which do not correspond well to the beginning and end of the template. Thus this technique can allow correct alignment of input with the correct-word template when non-speech intervals have been erroneously included at the beginning and end of the input word

during the endpoint detection. This endpoint adjustment option was introduced to alleviate the effect of unreliable endpoint detection, and was not used in later experiments after the endpoint detection had been improved. It can compensate for the inclusion of non-speech in the input, or the omission of initial and final parts of words in the templates, but not for omissions from the input or the inclusion of extraneous sounds in the templates: thus it is best suited for use in conjunction with an endpoint detection technique (applied to the input utterances to be recognised) which allows additional regions to be included beyond the probable word endpoints.

The other DTW parameters have been kept fixed during the experiments conducted with the system. On the basis of the results of preliminary experiments reported in section 4.2.4, no global constraints were adopted to restrict the locus for time registration paths within the parallelogram defined by the maximum and minimum slopes, and the absolute value metric was used in preference to the more computationally expensive squared Euclidean metric. The accumulated distance thresholds were useful for reducing the computation during one-stage experiments by allowing early abandonment of poorly-matching templates, but were judged to be unnecessary in multiple-stage operation, as a rather similar effect (though with a relative distance criterion instead of an absolute one) could be attained by the rejection of templates at the early stages, so that detailed matching would not even begin for unlikely candidate templates. Therefore the thresholds were set to fairly large values which would allow any plausible candidate to be matched in full.

The next section of this chapter describes experiments conducted with the data base already described, and with further data from a different set of speakers, to investigate the tradeoffs between speed and accuracy of recognition attainable with the multiple-stage system. Recognition accuracies and

computation times are given for various choices of the number of stages in the recognition procedure, the segmentation parameters at each stage and the decision threshold values.

4.4: Multiple-stage recognition experiments

4.4.1: Aims and design of experiments

If optimal performance is to be obtained from a multiple-stage recognition system as described in the preceding section, appropriate settings of a number of system parameters must be found. These include the number of stages to be used; the segmentation, number of segments and segment representation at each stage; and the threshold for template elimination after each non-final stage. Experiments were devised to explore the recognition accuracies and computational requirements resulting from various values of these parameters.

Several combinations of three or four sets of segmentation parameters (one set for each stage) were defined. In each case, a small number of segments per word was adopted at the first stage, and progressively larger numbers were used at later stages. The segment representation technique for each stage was determined on the basis of the findings reported in section 4.2.7.3 above: thus averaging was adopted at the first stage, and interpolation at the final stage, and averaging or interpolation was applied at each intermediate stage. Various combinations of linear segmentation and trace segmentation were employed.

For each of the combinations of segmentation parameters formed in this way, the recognition error rate and the average computation time per input word were found for each of a range of sets of template elimination threshold values. The error rate was plotted against the computation time for each set of threshold values, to give an indication of the accuracy-speed tradeoffs attainable

with the specified combination of segmentation parameters. The reduced cases in which not all the stages were used were plotted as extreme points: these cases correspond to the setting of the appropriate template elimination threshold values to 1.0 (to pass no templates on, so that only the earlier stages are used) or ∞ (to pass all templates on to the later stages, effectively eliminating the previous stage). (However, where a threshold value was set to ∞, the computation time was obtained by actually omitting the preceding stage, so as not to require the templates to be processed at that stage when they were in any case going to be passed on for matching at the following stage.)

Some of the experiments were conducted using a simulation mode, in which the word distances at each stage, for all combinations of input and reference utterances, were computed once and stored, and then these precomputed distances were used repeatedly in the experiments with different elimination threshold values. For each set of threshold values, and for each stage of the recognition procedure, the number of input utterances requiring use of that stage and the total number of word matching operations executed were recorded. These counts were multiplied by estimates of the times required per input word (for segmentation and associated overheads) and per matching operation, derived from previously conducted timing experiments, to obtain overall times per recognition.

4.4.2: Details of multiple-stage experiments

Multiple-stage recognition and accuracy experiments, using the procedures outlined in section 4.4.1, were conducted on three data bases: (1) the GP words and digits from the data base already described, produced by three speakers and represented by mel cepstral coefficients; (2) utterances of the digits by four

speakers, represented by linear predictive cepstral coefficients; and (3) words from a 50-word vocabulary spoken by one male speaker, also represented by linear predictive cepstral coefficients.

The details of the second data base were as follows. There were four speakers, two male and two female (of whom one of the males was the same as speaker 1 in the previous data base). For each of these speakers, two sets of templates were created, and 30 repetitions of the 10 digits were collected during several recognition sessions on separate occasions (usually 50 words per session, comprising five repetitions of each digit) using the interactive recognition system described in chapter 5 below. (More details of the procedures employed in the data collection may be found in chapter 6.) One of the two template sets for each speaker consisted of single-token templates (as in the segmentation experiments reported above); the other consisted of templates constructed by a robust two-token averaging procedure [30,31,255]. The acoustic representation of each word consisted of 12 cepstral coefficients per frame, derived by 24th-order LPC analysis. The frame shift was 10ms, and the frame length 25.6ms; a Hamming window was applied. (The order of the analysis was chosen to match the bandwidth of the input signal, which was lowpass filtered at 8kHz and sampled at 20kHz.)

The vocabulary for the third data base consisted of numbers up to "million", days of the week and month names, as listed in table 4.4. Six repetitions of the vocabulary (300 utterances in all) were collected from a single speaker (the same as speaker 1 in the previous speaker sets) and were recognised using each of two sets of templates. Again, one set consisted of single-token templates, and the other of two-token averaged templates. The data collection procedure and acoustic analysis were the same as for data base 2.

Table 4.4: 50-word vocabulary

zero	ten	twenty	million	March
one	eleven	thirty	Sunday	April
two	twelve	forty	Monday	May
three	thirteen	fifty	Tuesday	June
four	fourteen	sixty	Wednesday	July
five	fifteen	seventy	Thursday	August
six	sixteen	eighty	Friday	September
seven	seventeen	ninety	Saturday	October
eight	eighteen	hundred	January	November
nine	nineteen	thousand	February	December

For each of the three data bases, error rates and times were obtained as described above for each speaker and template set, and the results for each combination of segmentation parameters and elimination thresholds were averaged across the speakers and template sets to obtain overall results. Separate averaged results for the single-token template sets and for the two-token template sets were also computed in the case of the second data base.

The combinations of segmentation parameters considered in each case are listed in table 4.5.

4.4.3: Results

The results of the multiple-stage recognition experiments were plotted for each of the combinations of segmentation parameters as listed in table 4.5. Plots of error rate against time per recognition for selected cases are reproduced here as figures 4.7-4.20 (as indicated in the final column of table 4.5). (The results for the other combinations of segmentations were qualitatively similar.)

Figures 4.7-4.15 show results for the first data base – for the GP vocabulary with a variety of combinations of segmentation parameters (figures 4.7-

Table 4.5: combinations of segmentation parameters
(given as "segmentation, segments per word, segment representation")

Segmentation: T - trace segmentation; L - linear time segmentation
Representation: a - averaging; i - interpolation

Data base (vocabulary)	Segmentation parameters stage 1	stage 2	stage 3	stage 4	Figure numbers
1 (GP)	L 2 a	T 5 a	T 20 a	-	-
	L 2 a	T 5 a	T 30 i	-	4.7
	L 2 a	T 5 a	T 40 i	-	-
	L 2 a	L 10 a	L 20 a	-	4.8
	L 2 a	T 10 a	T 20 a	-	-
	L 2 a	L 10 a	T 30 i	-	-
	L 2 a	T 10 a	T 30 i	-	4.9
	T 2 a	T 10 a	T 30 i	-	4.10
	L 2 a	T 10 a	T 40 i	-	4.11
	T 2 a	T 15 a	T 30 i	-	-
	L 2 a	T 15 a	T 40 i	-	4.12
	T 2 a	T 15 a	T 40 i	-	-
	L 2 a	T 5 a	T 10 a	T 30 i	-
	L 2 a	T 5 a	T 10 a	T 40 i	4.13,4.14
	L 2 a	T 5 a	T 20 a	T 40 i	-
1 (digits)	L 2 a	L 10 a	L 30 i	-	4.15
2	L 2 a	L 10 a	L 29 i	-	4.16-4.18
3	L 2 a	L 10 a	L 29 i	-	4.19,4.20

4.14), and for the digits with one combination of segmentation parameters (figure 4.15). For each case with three sets of segmentation parameters, the times and error rates are plotted for each of a number of values of the threshold t_1 for template elimination after the first stage. The points on the graph for each value of t_1 correspond to different values of t_2, at intervals of 0.05, starting with 1.00. Thus, the first point plotted on each line represents the result with only the first and second stages in use. The last point on each line corresponds to $t_2 = \infty$ – that is, it represents the result with only the first and third stages. Where there are four stages (in figures 4.13 and 4.14), the results plotted in each diagram are for one value of t_1, and each line corresponds to a value of t_2, with a point for each value of t_3 (starting from 1.00); in this case the first point on each line represents a result obtained with only the first three stages.

(It should be noted that the scales on the axes are not the same for all of figures 4.7-4.20, and in particular that in figures 4.7-4.15 the error rate scale starts from 10%, not from 0.)

For comparison, the times for recognition of this data base using only the final stage (i.e. those corresponding to threshold values $t_n = \infty$ for all non-final stages n) were about 2.9s per input utterance for the digits vocabulary, or 5.9s for the GP words, with 20 segments per word; 6.7s or 13.5s with 30 segments per word; and 11.5s or 23.1s with 40 segments per word. (In each case, the time per (single-stage) recognition is roughly proportional to the size of the vocabulary.) The average time per recognition with no segmentation was 7.2s for the digits, or 25.1s for the GP words: in this case the time per recognition depends not only on the vocabulary size but also on the durations of the training and input utterances (which are typically greater for GP words than for digits).

After the experiments on this data base (and some other experiments with utterances of the digits), the segmentation parameters for use in subsequent work were chosen. The number of stages of comparison was set to be three, with linear time segmentation at each stage; the numbers of segments per word were fixed at 2, 10 and 29 at the respective stages; and the averaging technique was adopted at the first two stages, with interpolation at the third stage. These parameter choices result in 2, 10 and 30 vectors in the three segmented forms of each word. The choice of these parameters is explained in section 4.4.4.2 below.

Figures 4.16-4.18 show the results for the digits in data base 2, with the single-token and two-token template sets, and averaged over all the template sets. In each case, the previously determined segmentation parameters were used. The experiments with this data base were run on a different computer, a Masscomp MC5700, which was faster than the MC550 used for the experiments with the other data bases. To compare the results on the different data bases,

the times per recognition for data base 2 should be multiplied by 6.7. The time per recognition using only the third stage (30 vectors per word) was 1.29s. (This corresponds to the right-hand end of the line for $t_1 = \infty$ in each diagram.)

Figures 4.19-4.20 show results similarly obtained (on the MC550) for data base 3 (the 50-word vocabulary spoken by one speaker). Experiments were also conducted on this data base, with the single-token templates, using trace segmentation instead of linear time segmentation in some or all of the stages; the results were very similar to those with linear segmentation. The time per recognition using only the third stage was 49.6s.

4.4.4: Discussion of multiple-stage recognition results

4.4.4.1: Combinations of segmentation parameters

Figures 4.7-4.14 illustrate some general findings as to the effects of using different combinations of segmentation parameters in a multiple-stage recognition system. The criterion for evaluation and comparison of combinations of parameters is the best accuracy obtainable for a given amount of computation per recognition. For any specified computation time per recognition, this optimal accuracy can be estimated from the diagram by reading off the error rate from the plotted line which is lowest at the specified position on the computation time scale. For small values of computation time per recognition, the error rate should be read from a line (not shown) joining the initial points of the successive plotted lines: this corresponds to interpolating between the values of t_1 (or t_2 for four-stage recognition) for which results have been obtained. (The graph of optimal accuracy against time per recognition is the envelope of the (theoretically infinite) set of lines corresponding to all possible choices of t_1 – or choices of t_1 and t_2 in the case of four-stage recognition.)

Figure 4.7: results for three-stage recognition of GP words (parameters L 2 a; T 5 a; T 30 i)

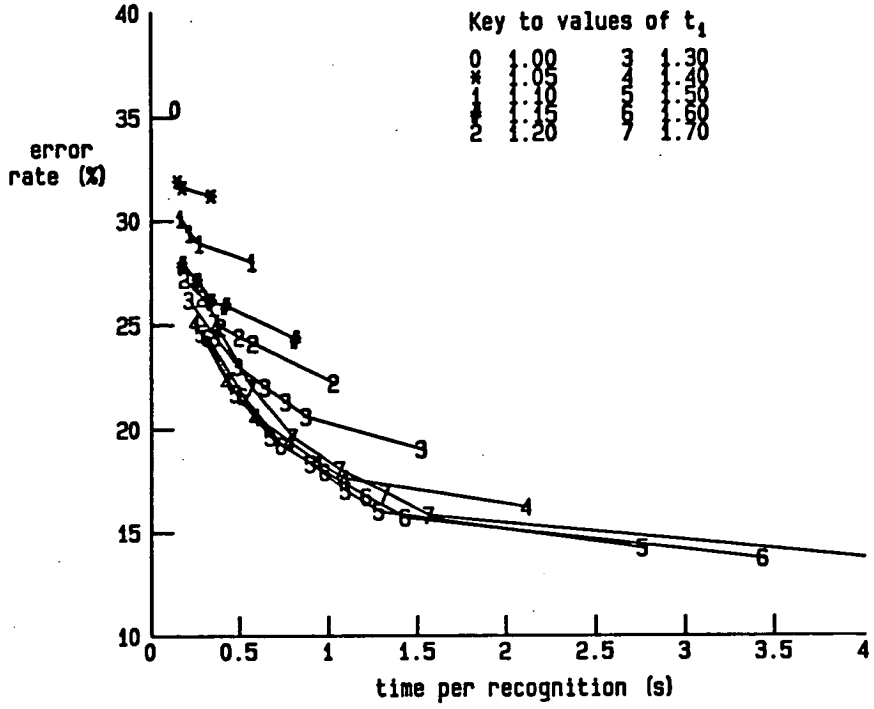


Figure 4.8: results for three-stage recognition of GP words (parameters L 2 a; L 10 a; L 20 a)

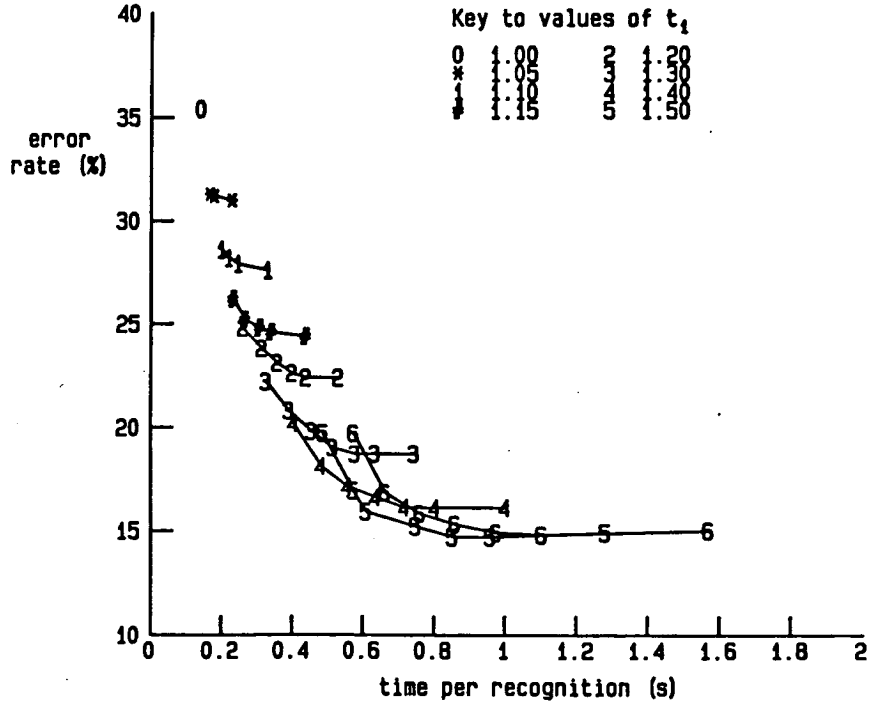


Figure 4.9: results for three-stage recognition of GP words (parameters L 2 a; T 10 a; T 30 i)

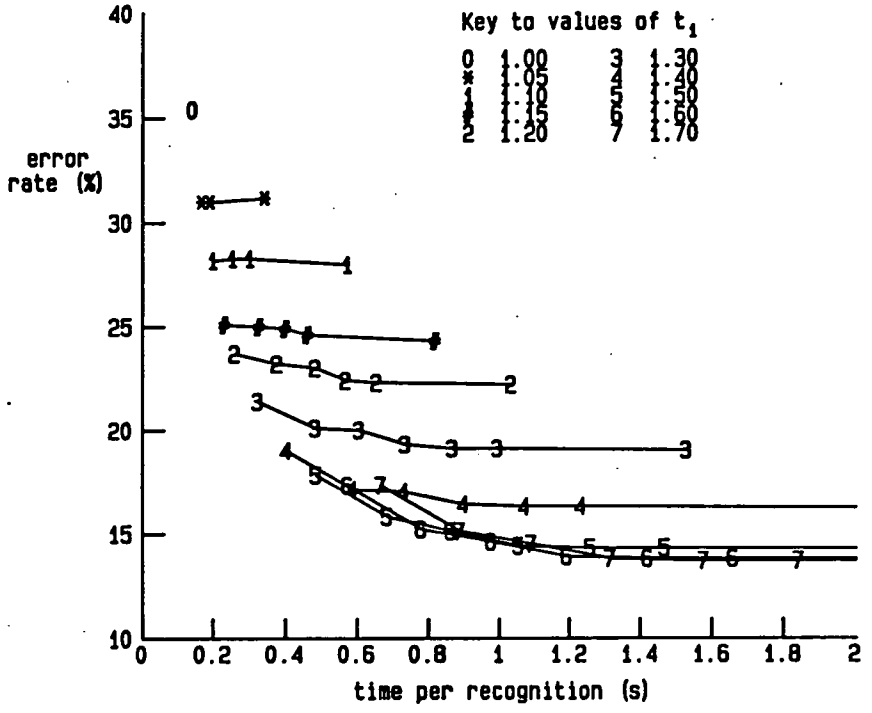


Figure 4.10: results for three-stage recognition of GP words (parameters T 2 a; T 10 a; T 30 i)

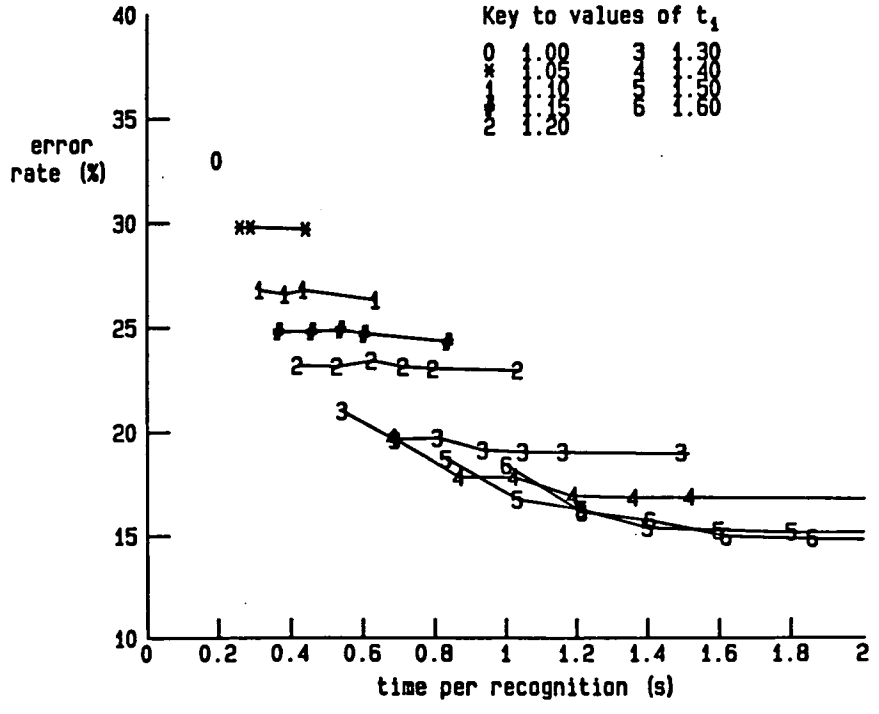


Figure 4.11: results for three-stage recognition of GP words (parameters L 2 a; T 10 a; T 40 i)

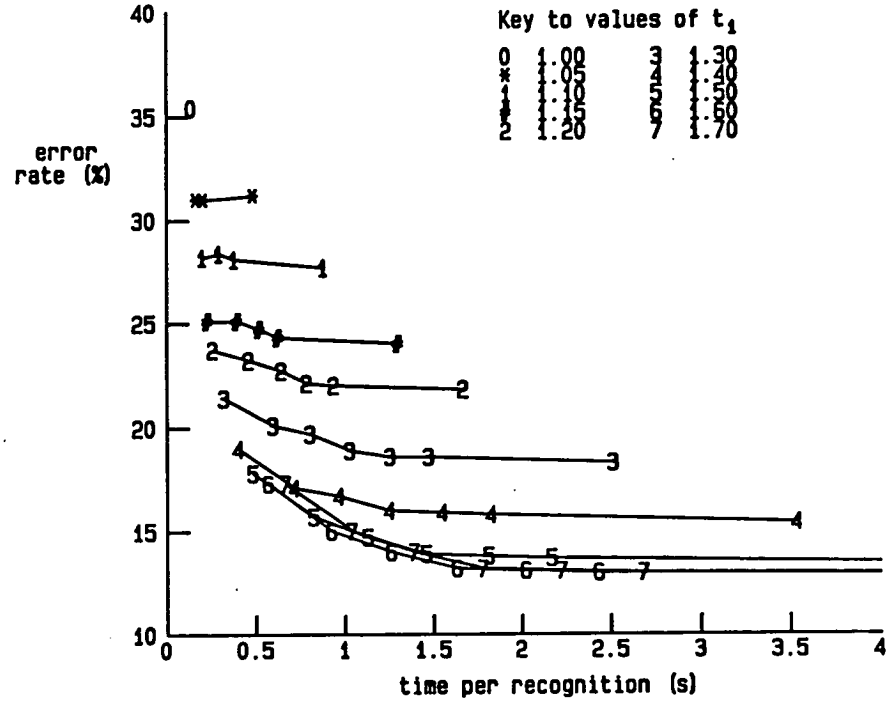


Figure 4.12: results for three-stage recognition of GP words (parameters L 2 a; T 15 a; T 40 i)

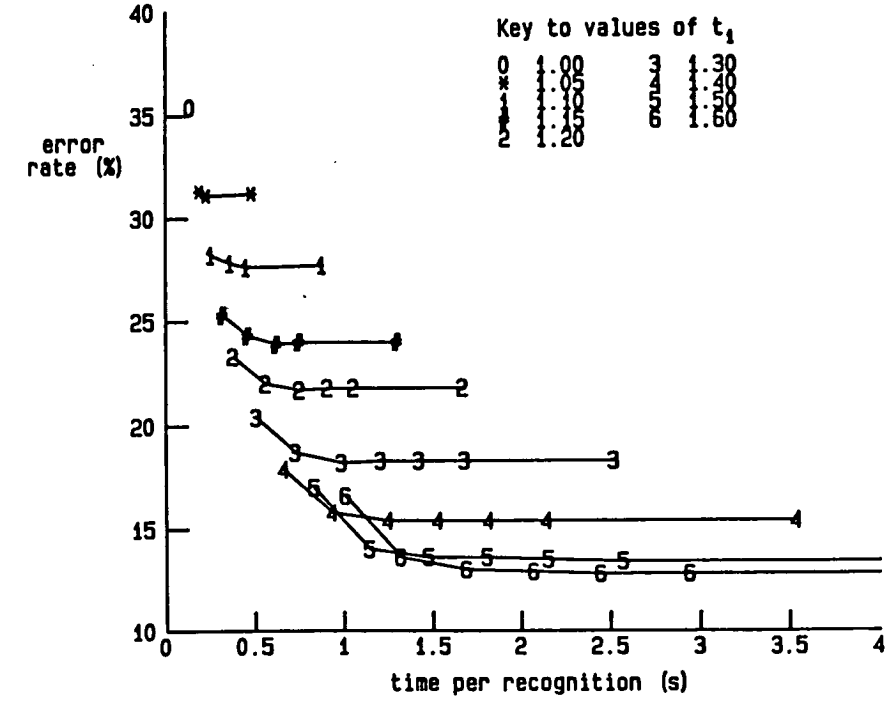


Figure 4.13: results for four-stage recognition of GP words (parameters L 2 a; T 5 a; T 10 a; T 40 i; t_1 = 1.5)

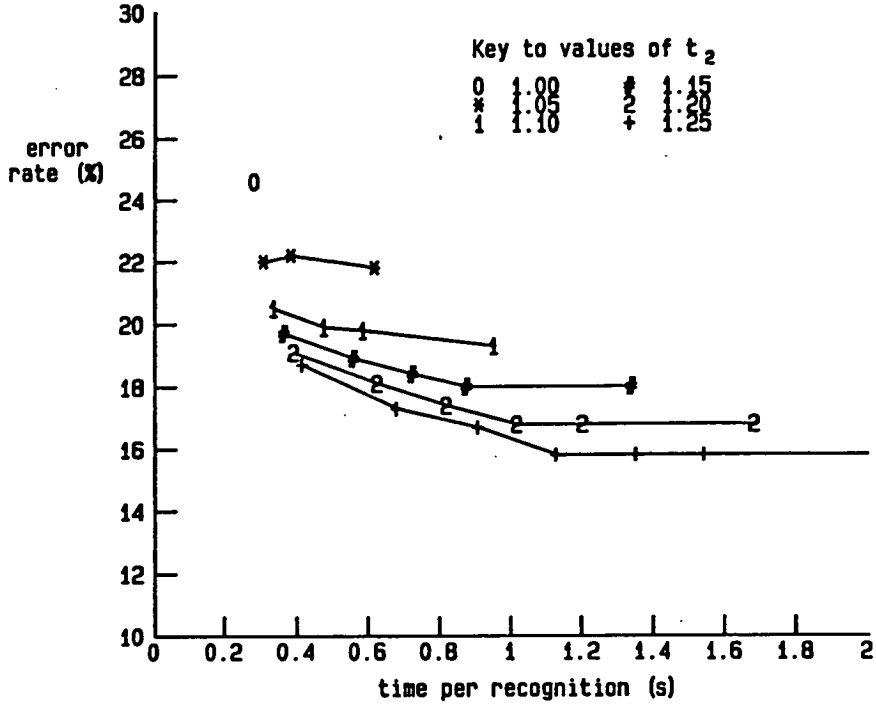


Figure 4.14: results for four-stage recognition of GP words (parameters L 2 a; T 5 a; T 10 a; T 40 i; t_1 = 1.6)

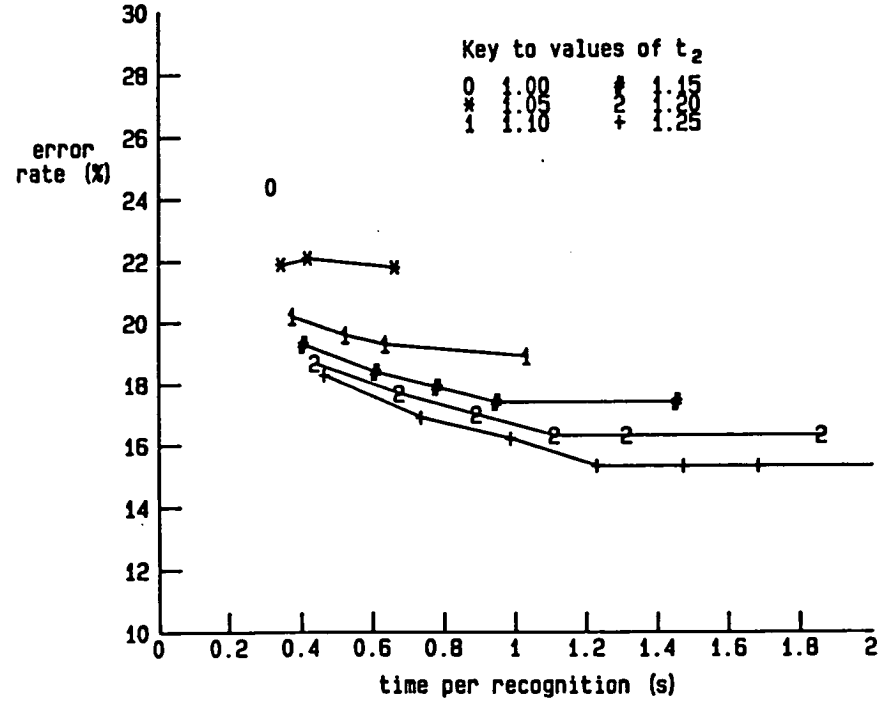
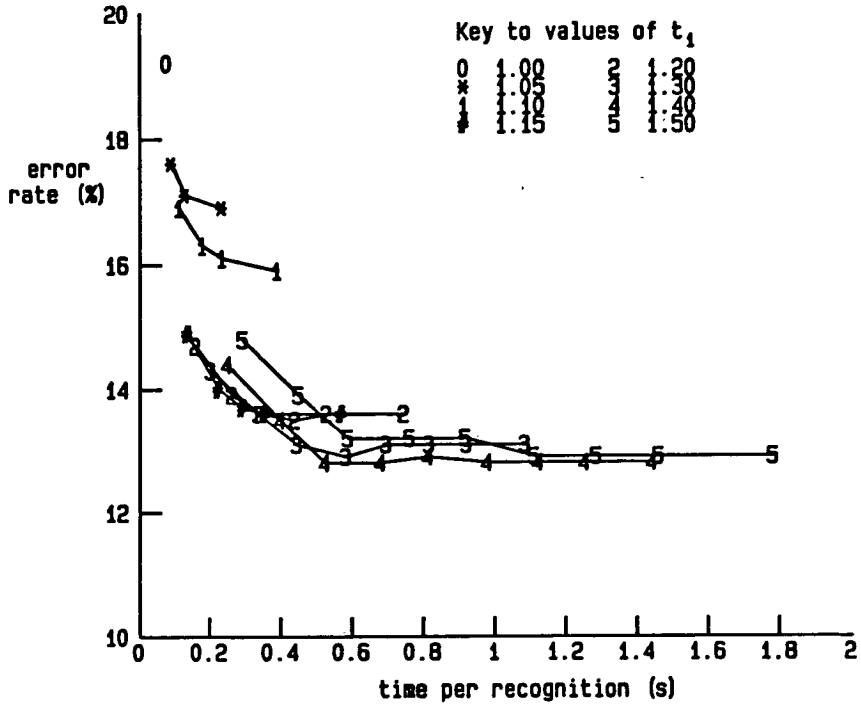


Figure 4.15: results for three-stage recognition of digits (data base 1) (parameters L 2 a; L 10 a; L 30 i)



In every case considered, the number of segments per word at the first stage was 2. With each word thus represented by two averaged vectors, the DTW matching at the first stage reduces to a linear alignment requiring only two vector distance computations. Thus the total computation required to recognise a word can be greatly reduced if many templates are eliminated after the first stage. The appropriateness of using such a simple first-stage comparison is confirmed by the observation that, once t_1 has been increased to 1.6, a further increase to 1.7 (allowing more templates to be retained for consideration at the second and subsequent stages), with no change in the value of t_2, yields at best a marginal improvement in recognition accuracy (figure 4.9), and in some cases actually causes more errors (figure 4.7), while increasing the computation time.

Also, it appears, from comparison of the results in figures 4.13 and 4.14 with the corresponding results ($t_1 = 1.5$ or 1.6) in figure 4.11, that inserting a second early elimination stage, with five segments per word, only increases the computation time required to attain a specified accuracy: the use of a five-segment comparison does not result in more effective elimination of templates than can be achieved with the two-segment comparison alone. (No results with $t_2 > 1.25$ are shown in figures 4.13 and 4.14; but the use of a large value of t_2 can be expected to yield almost the same results as with the second stage omitted, as in figure 4.11 – with some increase in computation time.) The effect of omitting the first stage completely (or setting t_1 to ∞) can be seen in the results for recognition of data bases 2 and 3, plotted in figures 4.16-4.20: the best accuracy attained using the remaining two stages is similar to the best accuracy with the three-stage configuration and a t_1 value of 1.6 – or worse with the digits and single-token templates (figure 4.16) – while the computation time per recognition is several times greater.

In the experiments with data base 1, the best accuracy with a computation time under 2s per recognition (error rate 13.0%, with computation time 1.7s) was attained using 2, 15 and 40 segments per word (figure 4.12). This is almost as good as the 87.2% accuracy (12.8% error rate) obtained in one-stage recognition with no segmentation. The three-stage result is 0.9% poorer than the accuracy obtained using only the final stage (40 segments per word and interpolation, resulting in 41 vectors per word), which is plotted in figure 4.4.

The accuracy attainable was very similar (13.2% error rate at 1.6s per recognition), or better if rapid computation (under 1s per word) was required, when the number of segments at the second stage was reduced from 15 to 10 (figure 4.11). A comparison of results with 5 and 10 segments at the second stage (as in figures 4.7 and 4.9 respectively) indicates that better accuracy for a

Figure 4.16: results for three-stage recognition of digits
(data base 2; single-token templates) (parameters L 2 a; L 10 a; L 29 i)

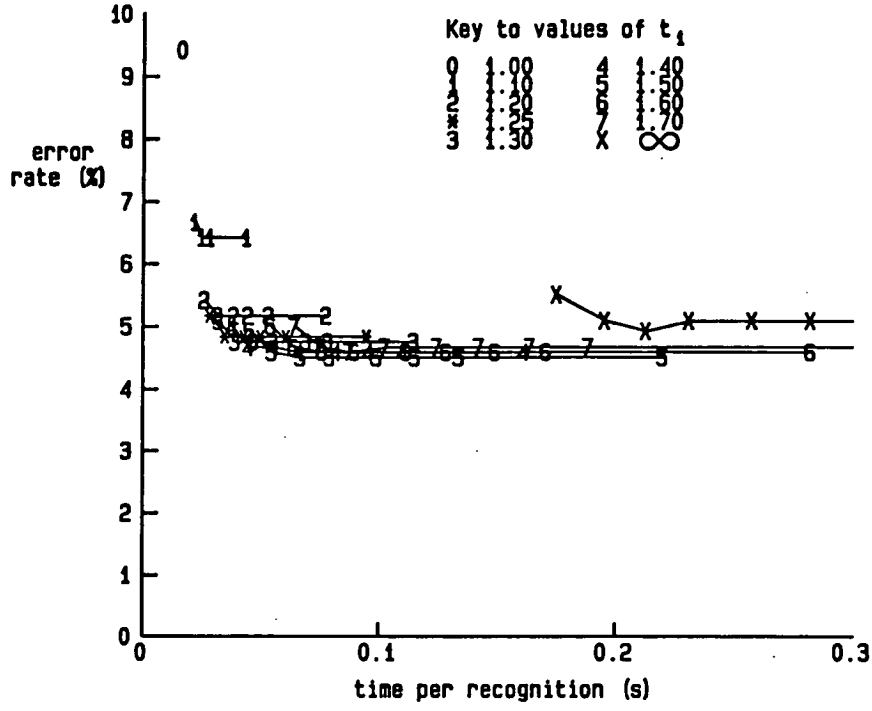


Figure 4.17: results for three-stage recognition of digits
(data base 2; two-token templates) (parameters L 2 a; L 10 a; L 29 i)

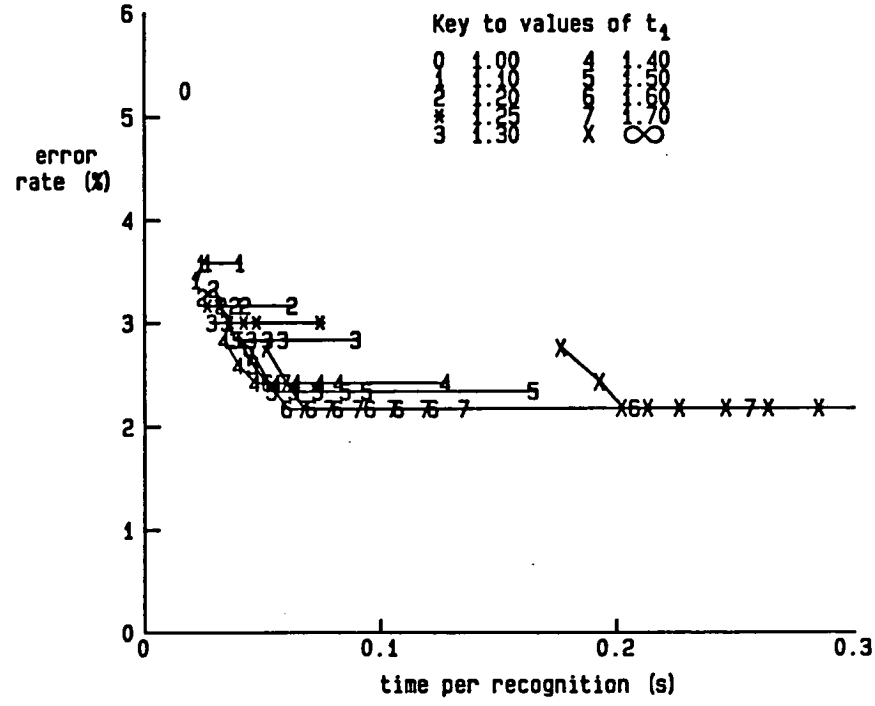
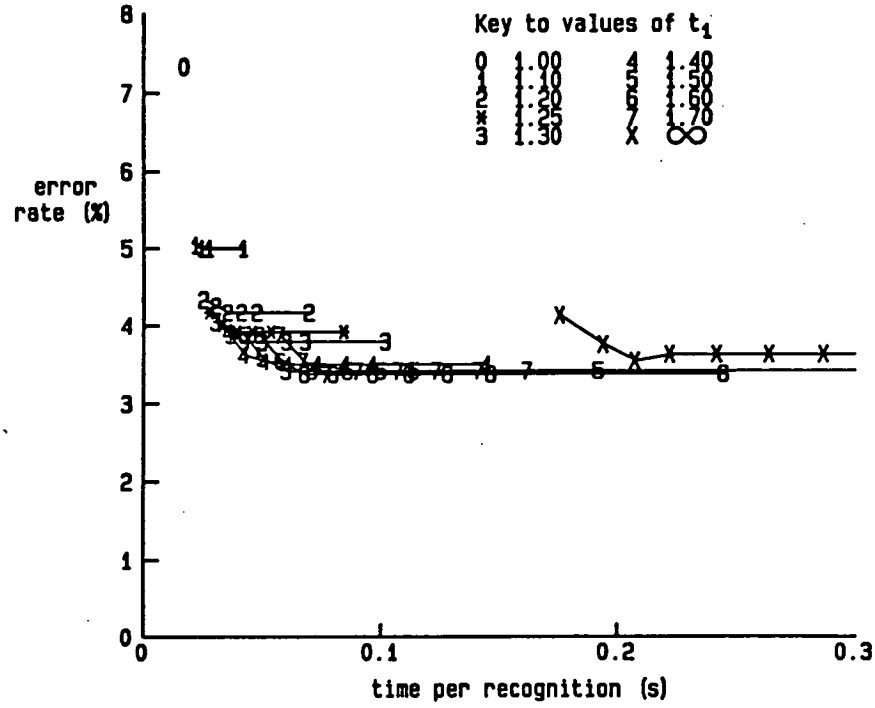


Figure 4.18: results for three-stage recognition of digits
(data base 2; averaged over template sets) (parameters L 2 a; L 10 a; L 29 i)



given computation time can be attained using 10 than using 5. Although no three-stage experiments were conducted with more than 15 segments per word at the second stage, a comparison of the four-stage results with 10 segments at the third stage (as shown in figures 4.13 and 4.14) and with 20 shows that, in this case, using a larger number of segments per word does not improve the tradeoff of speed and accuracy; it is to be expected that this result would still hold true if the preceding stage (using 5 segments per word) were omitted. Thus the optimal number of segments per word at the second stage appears to be in the region of 10 or 15.

The results with 2, 10 and 30 segments per word at the three stages (figure 4.9) show better performance for computation times under 1.4s per recognition than when 40 segments per word were used at the third stage (figure 4.11).

Figure 4.19: results for three-stage recognition of 50-word vocabulary (single-token templates) (parameters L 2 a; L 10 a; L 29 i)

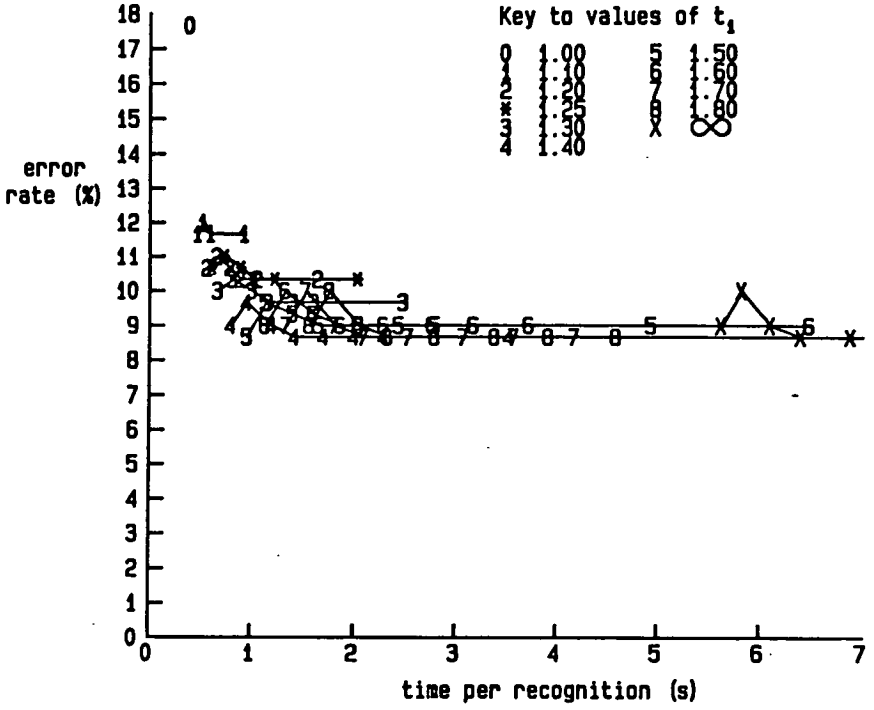
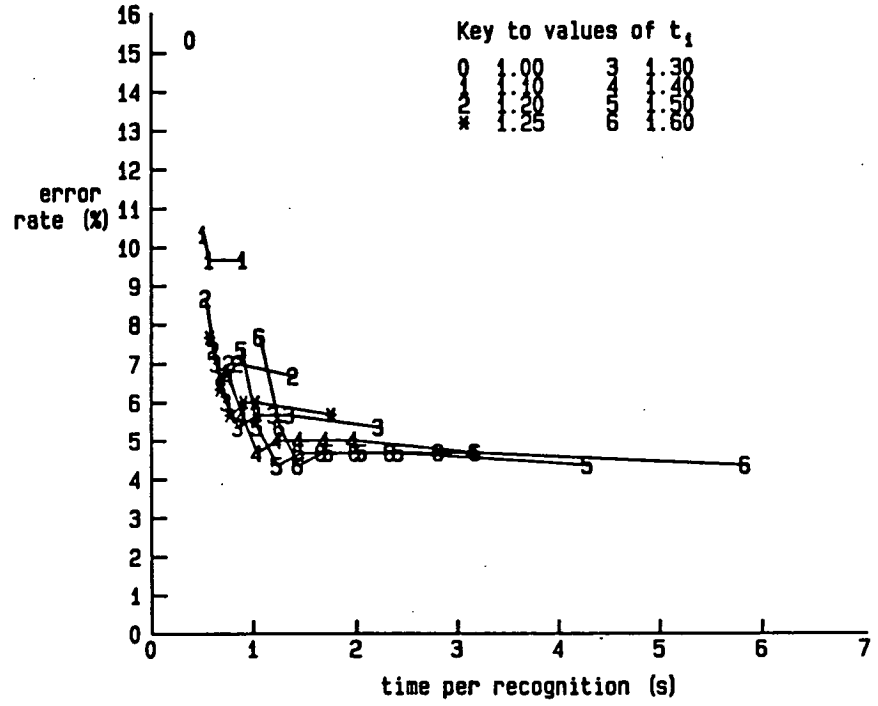


Figure 4.20: results for three-stage recognition of 50-word vocabulary (two-token templates) (parameters L 2 a; L 10 a; L 29 i)



With 30 segments at the third stage, however, very little further improvement was attained when more computation time was allowed, and the lowest error rate was 13.7% instead of 13.2%. Comparison of the results in figure 4.8 with those for the combination "L 2 a; L 10 a; T 30 i" (as in figure 4.9 but with linear time segmentation instead of trace segmentation at the second stage) shows a similar pattern when the number of segments at the third stage is reduced from 30 to 20: the results with 20 were better for computation times under 1.0s, but the lowest error rate attained overall was 14.6%, whereas with 30 segments per word at the final stage the error rate could be reduced to 13.4%.

The results for the GP words with 2, 10 and 30 segments per word show the effects of different combinations of segmentation techniques. In each case trace segmentation was applied at the third stage; but in one case (not plotted here, due to accidental loss of the results file) the first two stages used linear time segmentation; in the second case (figure 4.9) trace segmentation was introduced at the second stage; and in the third case (figure 4.10) trace segmentation was used throughout.

A comparison of the results for the first two of these cases shows that when only the first two stages were used (i.e. with t_2 set to 1.0 – as represented by the first point plotted for each value of t_1) the use of trace segmentation at the second stage (figure 4.9) gave markedly better recognition accuracy than with linear time segmentation (which was what might be expected from the results with 10 vectors per word in figures 4.3 and 4.4), but as the value of t_2 was increased, to allow more use of the third stage, the results with linear segmentation at the second stage became similar to the corresponding results with trace segmentation, or slightly better. A possible explanation is that it is beneficial to have different segmentation techniques at the second and third stages, so that, if the decision between two candidate templates can be made reliably when they

are compared with the input using the first type of segmentation, the decision is taken at the second stage, but if not, they are passed on to the third stage where the other segmentation is applied. There may be some words which are more reliably recognised when linearly segmented, and others for which trace segmentation is more effective.

The comparison of figures 4.9 and 4.10 reveals a similar effect with the choice of linear segmentation or trace segmentation at the first stage, when the second stage incorporates trace segmentation. For small values of t_1, the error rates with trace segmentation at the first stage, shown in figure 4.10, were lower than those with linear time segmentation shown at the corresponding points in figure 4.9; but for larger values of t_1 the results with linear segmentation at the first stage were better. Another difference apparent from these figures is that the computation time is increased by using trace segmentation at the first stage, even when (as happens when $t_1 = 1.0$) the number of templates passed on to the second stage is unaltered. This is because of the vector distance calculations required to measure the trace during the segmentation of each input word.

4.4.4.2: Choice of segmentation parameters

It is clear from the results discussed above that a three-stage recognition configuration, with 2 segments per word at the first stage, 10 or 15 at the second stage, and 30 or 40 at the third, yields near-optimal accuracy and speed of recognition. From the previous experiments comparing segment representation techniques, it is evident that the vectors should be derived by averaging at the first and second stages, and by interpolation at the final stage. On the basis of these results, and vocabulary considerations, a decision was made as to the system parameters to be adopted for subsequent experiments.

As most of the experiments to be carried out with the multiple-stage recognition system were with the digits vocabulary, on which better results had been obtained with linear time segmentation than with trace segmentation, linear segmentation was adopted for use at each of the three stages. (If the explanation suggested above for the results on the GP vocabulary with different combinations of linear segmentation and trace segmentation is correct, there might have been some advantage in using trace segmentation instead of linear segmentation at the second stage for the digits; but, from a comparison of the results in figure 4.9 with the corresponding results with linear segmentation at the first two stages, it seems likely that the advantage, if any, would be only a slight one.)

Also on account of the vocabulary, the numbers of segments per word were fixed at 2, 10 and 29 (yielding 2, 10 and 30 vectors per word respectively): it seemed inappropriate to use 40 segments per word when the words to be represented were mostly monosyllabic and of short duration (often having fewer than 40 frames), as the digits are.

Some further investigations of the effects of the segmentation technique were later carried out with the 50-word vocabulary; the replacement of linear segmentation by trace segmentation at some or all of the three stages gave performance indistinguishable from that obtained with linear segmentation throughout.

4.4.4.3: Template elimination threshold values

As already mentioned (in section 4.4.4.1), near-optimal recognition accuracy for a moderate amount of computation can be reached when the value of the threshold t_1 for template elimination after the first stage is approximately 1.6.

The smallest value of t_2 at which the best possible accuracy is attained, when t_1 = 1.6, ranges from 1.1 (as in figure 4.17) to 1.2 (figure 4.9). Similar accuracies, with reduced computation, can be attained with t_1 equal to 1.5 and the same value of t_2. The results on the digits from data base 1 (figure 4.15) and on data bases 2 and 3 (figures 4.16-4.20) show that reducing t_1 to 1.5 may even improve the accuracy (figure 4.16), and on these data bases a further reduction of t_1 to 1.4 degrades the accuracy only slightly or not at all. The further reduction of computation time is best achieved by reducing t_2 to a value close to 1.0, while keeping t_1 at a constant value: reducing t_1 below about 1.4 is appropriate only if the speed (rather than accuracy) of recognition is the main consideration.

The choice of the most appropriate elimination threshold values for any particular application of the multiple-stage recognition system will depend on details of the implementation and of the application. Relevant considerations include the size and difficulty of the vocabulary to be recognised; the number of templates per word; the processing power of the machine on which the recognition system is implemented (and, if it is a time-sharing system, the level of competition from other processes running simultaneously); and the balance of speed and accuracy requirements for the desired application. For any given specification of speed and accuracy requirements, the most suitable settings of t_1 and t_2 can be determined from the plot of error rates against computation times (with the times normalised, if necessary, for machine power and load) for the requisite vocabulary.

For the application primarily in view here, namely research into recognition accuracy improvements attainable through template adaptation, the first requirement was near-optimal accuracy, to allow realistic simulation of the performance of a hardware recognition system using customised components, in which speed would be less of a problem than with a software system on a

general-purpose computer.

In view of the results on data base 1, and some preliminary results on digits represented by linear predictive cepstral coefficients (later found to have been incorrectly computed, and therefore replaced by data base 2), the threshold values t_1 and t_2 were set at 1.6 and 1.2. These threshold values were used in the interactive recognition sessions in which data bases 2 and 3, and additional data for subsequent experiments, were collected, and also in some of the later experiments as described in chapters 5 and 6. Other pairs of template elimination threshold values (t_1,t_2) adopted for some later experiments (chapters 6 and 7) were (1.4,1.15), (1.6,1.1) and (1.6,1.12).

With these segmentation parameters and threshold values, the average computation per recognition can be reduced by a factor of about 20 in the case of the digits vocabulary (figures 4.16-4.18), or about 30 in the case of the 50-word vocabulary (figures 4.19-4.20), without any loss of accuracy, relative to the case where the final stage of comparison is applied to all templates. It should be remembered that this is an average: with the three-stage comparison, the computation time varies from one recognition to another, according to the difficulty of recognition of particular input utterances, whereas with single-stage recognition (assuming no early abandonment of template matches by accumulated distance thresholding) the time per recognition will be nearly constant. (These results on data bases 2 and 3 show greater improvements in efficiency than the results on data base 1, where the computation reduction factor is only about 15 for the digits. This confirms the finding that data base 1 is of poorer quality in some respect (perhaps because of suboptimal endpoint location or acoustic processing), as indicated by the generally high recognition error rates on this data base.)

These results are for unadapted templates. When the templates have been optimised by adaptation, even greater computational efficiency can be attained (as reported in sections 6.3.2.5 and 7.3.3.4).

4.5: Summary of results

In this chapter, some experiments with time segmentation and segment representation techniques, as preprocessing for isolated word recognition using DTW, have been described, and the results have been presented and discussed. Two segmentation techniques – trace segmentation and linear time segmentation – were compared; for one vocabulary (the digits), linear time segmentation yielded slightly better results than trace segmentation, while for another vocabulary (the "golden passage" (GP) vocabulary) trace segmentation showed a slight advantage over the linear segmentation. Once a word has been segmented, using either of these techniques, representations must be derived for the segments (or segment boundaries). It was found to be best to average the acoustic parameter vectors in each segment when the number of segments per word was small (i.e. less than about half the average number of vectors in an unsegmented word pattern), and to interpolate vectors at segment boundaries when the number of segments per word was large.

The recognition results obtained with appropriate segmentation and interpolation, with a number of segments per word approaching the typical number of frames per word before segmentation, were slightly better than those with no segmentation applied. Also, results only a little poorer than those with many vectors per word were obtained – with a considerable saving in computation in the DTW matching – when each word was represented by only a few vectors, each derived by averaging. These features of the results led to the idea of a

multiple-stage recognition system, with a computationally economical matching using a small number of segments per word at the first stage to eliminate the most poorly-matching templates from consideration, and increasingly detailed comparisons using larger numbers of segments per word at the subsequent stages to yield optimal accuracy in the final recognition. Such a system was constructed, and was tested to determine appropriate settings of its parameters.

The best combinations of accuracy and speed of recognition were found to occur when three stages of comparison were used, with approximately 2, 10 and 30 segments per word respectively. The results with the three-stage system show recognition accuracies similar to those attained using the final stage alone (or by single-stage recognition without segmentation), with reduction factors in the average computation required per recognition ranging from 15 to 30 across the data bases used in the experiments. In some cases, the use of the multiple-stage decision procedure yielded a slightly higher accuracy than was attained using the final stage alone.

The three-stage recognition procedure was adopted for use in the subsequent experiments (to be described in chapters 6 and 7) with template adaptation. This allowed more experiments to be completed in the time available than would have been possible without the substantial improvement in computational efficiency. It was also employed in the interactive recognition sessions (described in chapter 6) in which the main data base for speaker-dependent recognition experiments was collected; its use in these sessions helped to make the data collection procedure more realistic as a simulation of a practical application of speech recognition, by reducing (by several seconds) the recogniser's response time for each input utterance.

CHAPTER 5

AN INTERACTIVE WORD RECOGNITION SYSTEM WITH TEMPLATE ADAPTATION

5: AN INTERACTIVE WORD RECOGNITION SYSTEM WITH TEMPLATE ADAPTATION

5.1: Introduction

The isolated word recognition system described in section 4.3 has been further developed to operate interactively (using direct speech input, rather than requiring previously stored analysed data), and to incorporate adaptation of templates by weighted averaging with recognised utterances. The interactive system includes automatic endpoint detection and LPC analysis, as well as the actual recognition component. This chapter describes the modes of interaction and adaptation which have been implemented.

With this interactive adaptive recognition system, it is possible to explore experimentally the topics of adaptation by the system to the user's speech and of more general user-system interaction and interface design, which were identified in chapter 3 as being of particular relevance to the development of accurate and readily usable speech recognisers. The interactive mode is useful in allowing demonstration and observation of the ways in which a user may respond to a recognition system (with a particular form of interface design) during a recognition session; it also permits more realistic data collection than would be achieved in a recording procedure without immediate recognition of the words, as the manner in which the user speaks when simply recording data may not be the same as when interacting with a recognition system.

The next two sections give details of the interactive mode and of the adaptation options, respectively. The final section of this chapter contains the results of some experiments conducted to determine appropriate settings of certain system parameters. Experiments and results with template adaptation, and obser-

vations on the interaction between the system and the user, are reported in the subsequent two chapters.

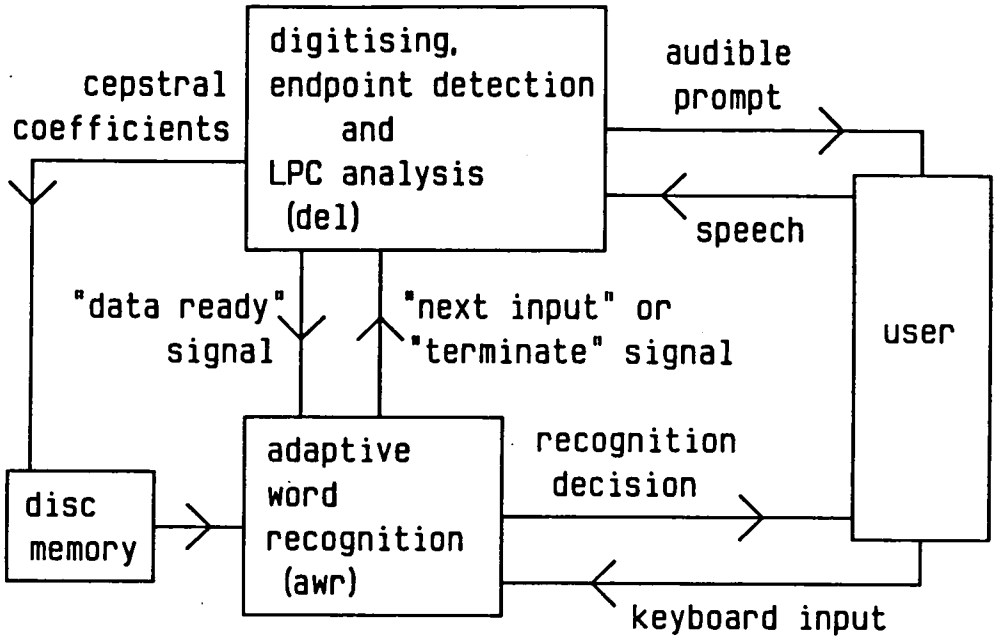
5.2: The interactive mode

To facilitate interactive use of the recognition system already developed, it was necessary to compose software to digitise the input, to detect the beginning and end of the word spoken and to perform some form of signal analysis in each time frame of the detected utterance; to interface this software to the recognition program; and to make provision for control of the system by the user (by speech or keyboard input) to allow the identification and correction of errors and the termination of the recognition session.

The digitising, endpoint detection and acoustic analysis were implemented in a C program (*del*) making use of the analogue-to-digital convertor and array processor on the MC550 computer. This program prompts the user to speak a word in a designated time interval (set to 1.5s for isolated word input), takes in the digitised waveform, performs endpoint detection and LPC analysis, and writes a sequence of vectors of acoustic parameter values to a file. Once this has been done, a signal is sent to the recognition program (called *awr*, for "adaptive word recognition"), which then reads the analysed data from the file and performs recognition. When the recognition decision has been made (and the user's response obtained, if verification is in use), *awr* signals *del* to prompt the user for another utterance, or to terminate if the user has indicated that the session is finished. The interactions of the system components with each other and with the user are shown in figure 5.1.

Prompting for speech input is by audible signals: the system gives a double "beep" (using the standard facility built into the terminal) at the start of the

Figure 5.1: interactions of user and recognition system components



recording interval, and a single "beep" to mark the end of it. The input is passed through a lowpass filter with a cutoff frequency of 8kHz, and the resulting waveform is digitised at 20kHz, with 12-bit resolution.

Once the input has been digitised, the endpoint detection processing begins. This involves computing the power of the signal and the number of zero-crossings in each successive 10ms interval, and then applying thresholds on the power and zero-crossing rate to find the frames at which the speech begins and ends. (The measure of signal power used in this system is not the true power, which is proportional to the sum of squares of sample values, but the sum of their absolute values.) The algorithm is a development of one devised elsewhere [46] in which provisional endpoints are determined using the signal power and then the word is extended at either end if a region of high zero-crossing rate is detected there (to allow the inclusion of certain low-power speech sounds such as voiceless fricatives).

Three power thresholds are used: inner and outer start/end thresholds, to find the points of transition between background noise and speech, and a threshold on the maximum power attained during a possible word, to determine whether it should be accepted. The provisional start frame is the first frame at which the outer threshold value is exceeded, or the first frame at which the inner threshold is exceeded if this occurs more than a specified number of frames later. When the start of a word has been detected, the powers in subsequent frames are examined until either the power has dropped below the outer threshold and remained below it for too many successive frames (in which case the provisional word is abandoned and the search for a new start frame begins) or else the power has exceeded the maximum power threshold for the required number of frames to confirm the detection of a word. If the detection is confirmed, the (provisional) end of the word is found using the inner and outer thresholds. These inner and outer thresholds are adapted dynamically, by a heuristically determined formula, according to the level of background noise, which is determined from the signal power in frames not classified as speech.

If two words are detected which are close together, they may be combined to form a single word, provided that the resulting word does not exceed a stipulated maximum duration. This is to allow correct detection of words containing low-energy intervals (e.g. in stop consonants). If, after the combining of any such closely adjacent words, a word has less than the specified minimum duration, it is rejected: this allows elimination of brief but loud non-speech noises.

After the detection of provisional endpoints by this procedure, the zero-crossing rates are examined in frames within specified intervals before and after each detected word. If a pulse of high-zero-crossing frames of sufficient duration is found, the word is extended to include this. Also, if the maximum zero-crossing rate attained in any high-zero-crossing pulse is more than twice the

threshold value, and the pulse is either close to the provisional start of the word (with no preceding zero-crossing pulse detected) or after the provisional end, it is included regardless of its duration. This latter case is permitted to improve the detection of word-initial and word-final stop consonants.

The endpoint detection parameter values used are listed in table 5.1. These were determined by experiments on 90 words from the data base of section 4.2.1, so as to optimise the correspondence of the automatically and manually located endpoints, and subsequently modified to improve the performance in interactive data collection. The power thresholds are expressed in arbitrarily scaled units; the time thresholds in 10ms frames; and the zero-crossing threshold in zero-crossings per frame. The "interval to search for zero-crossing pulses", as listed in table 5.1, is the length of the interval before the provisional start of the word in which the zero-crossings are examined; the length of the interval searched after the provisional end, however, is $t-1$ frames, where t is the length (in frames) of the minimum time between words: thus, with parameter settings such as those shown in the table, a longer interval is searched at the end of a word than at its beginning. A correction is made for DC offset (measured over the first 10 frames of the digitised signal) before the zero-crossing counts are computed.

The threshold adaptation rate is the quantity by which the value of $p(t)-f(P)$ is multiplied to obtain the increment in the power threshold P at frame t, where $p(t)$ is the power in this frame and f is a suitably chosen quadratic function (which is of the form

$$f(P) = aP^2 + bP, \tag{5.1}$$

with a and b positive, and thus is increasing over positive values of its

argument P). (The motivation for using a function of this form is that the threshold level P corresponding to a steady noise level p_0 – which will be such that $f(P) = p_0$, once the threshold adaptation has converged – should exceed p_0 by a larger factor when p_0 is small than when p_0 is large. The underlying assumption is that the background noise level may vary considerably from one input to another but the volume range of the word to be detected will remain relatively consistent: under these conditions it is desirable to keep the endpoint detection threshold well above the noise level so long as this does not cause it to exceed the typical word-initial or word-final speech power level.)

The endpoint detection parameter settings listed in table 5.1 were chosen specifically for the case where a single word is expected to be spoken during each digitisation interval. For the more general task of locating an indefinite number of utterances in each digitised interval, it might be appropriate to alter some of the parameters – for instance, to reduce the minimum time permitted between words, so as to prevent two words spoken in rapid succession from being counted as a single word.

Table 5.1: endpoint detection parameters

Parameter	Value
outer start/end threshold (before adaptation)	10.0
inner start/end threshold (before adaptation)	15.0
rise/fall time threshold	4
high power threshold	25.0
minimum number of high-power frames	4
maximum number of successive low-power frames	3
minimum word length	10
maximum word length	120
minimum time between words	40
threshold adaptation rate	0.1
interval to search for zero-crossing pulses	25
zero-crossing threshold	27
minimum duration of zero-crossing pulse	3

If (after all the adjustments described above) exactly one word has been detected in the digitised input, the LPC analysis is applied in each frame of the detected word. Otherwise, the system abandons the current input and prompts the user for a new utterance.

The LPC analysis is performed by the autocorrelation method, with a 25.6ms (512-point) Hamming window every 10ms. Preemphasis (factor 0.98) is applied to the speech prior to this analysis. Cepstral coefficients are derived from the linear prediction coefficients by a recursion formula [20,21]. The analysis order and the number of cepstral coefficients output per frame can be specified separately. (Experiments to determine appropriate values of these parameters are described in section 5.4.)

Once a word has been digitised, detected and analysed, the recognition process begins. The recognition program *awr* receives the "data ready" signal from *del*, reads the sequence of cepstral vectors, and performs segmentation, DTW comparison and template elimination as described in chapter 4. When the recognition decision has been reached, the identified word is written to the terminal screen, allowing the user to check whether the recognition is correct and respond accordingly. If no recognition decision can be taken, because the word distance ratio is below the rejection threshold, an asterisk is output instead; the user can then speak the word again until it is recognised by the system. The processing for one input utterance, from the beginning of the digitisation interval to the output of the recognition, takes typically about 10s on the MC550 for a small vocabulary (such as the 10 digits). Most of this time is occupied by the digitising, endpoint detection and LPC analysis. When the vocabulary is larger (or more confusable, so that more templates are matched at the later stages of the comparison process), the time for digitising, detection and analysis remains the same (if the average word duration is the same), but the time for the template

comparison is increased.

There are two main modes of interaction which can be selected by setting the system parameters. The first of these involves verification of each recognition by the user (through the keyboard); the second relies on the use of designated control words incorporated in the vocabulary, spoken and recognised in the usual way, to control the system's operation.

If the verification option is selected, the system waits, once the recognition has been output, for the user to respond by keyboard input. Possible user responses, and the system's actions on receiving them, are listed below. ("<return>" denotes a carriage return, and "<space>" denotes spacebar, on the keyboard.)

<return>	accept recognition as correct, and signal *del* to prompt for another utterance
<space><return>	recognition incorrect: delete recognition from screen; if this is the first recognition candidate, then find the second-best candidate and display it; otherwise, signal *del* to prompt for another utterance
/<return>	abandon utterance: delete recognition, and signal *del* to prompt for another utterance
q<return>	end recognition session (and signal *del* to terminate)
r<return>	retrain one or more templates: prompt the user for details, collect new training utterances (using *del*) and form a new template for each word specified; when retraining is complete, signal *del* to prompt for next recognition input

In cases where a recognition is indicated by the user to be correct or incorrect, this information is recorded in the results file, which also contains other details of each recognition such as the ratio of the best two word distances. If the utterance is abandoned (by typing of "/" or "q"), the system does not make any assumption about the correctness or incorrectness of the recognition. This allows the user to exclude from the subsequently computed recognition statistics any cases where noises in the background or mistakenly uttered words are detected and recognised.

The retraining procedure allows the user to select a word of the vocabulary for which a new template is to be formed, and to specify how many utterances are to be averaged together to form this template, and (if more than one utterance per template is specified) what threshold value should be imposed on the distance per frame as a precondition for averaging. Once the user has specified these details, the system prompts for an utterance of the word. If the template is to be formed from a single utterance, this utterance (once detected and analysed by *del*) is adopted as the new template which replaces the template previously in use for that word. (If the existing templates include more than one for the specified word of the vocabulary, the one that is replaced is the first of these in order of appearance in the template list.) Otherwise, the system continues to collect utterances, and compare them by DTW matching, until two are obtained whose average distance apart (per frame) is less than the stipulated threshold; it then averages these two together (again using DTW alignment: details of this process are given in section 5.3) to form the replacement template. If the specified number of utterances is greater than two, further utterances are collected and incorporated into the template (on meeting the threshold condition) until it is the average of the required number of utterances. (This is the same procedure [86] which is employed – in a separate training program – for the

formation of the initial template set.) Once the new template has been formed, the user has the option of specifying one or more further words of the vocabulary for retraining, or of returning immediately to the recognition session.

If verification by keyboard input is not in operation, the control of the system by the user depends on the incorporation of certain special words in the vocabulary. When one of these words is recognised during a recognition session, the system responds in an appropriate way. These control words can be used, like the keyboard inputs in the verification option, to identify misrecognitions, to initiate retraining or to terminate the recognition session. The control words permitted and the system's responses to them are as follows.

STOP	end recognition session (and signal *del* to terminate)
RETRAIN	retrain the template or templates for specified word or words (obtaining the details from the user by keyboard input, as above)
CORRECTION	previous recognition incorrect: signal *del* to prompt for another utterance

When an input is recognised as any word in the vocabulary other than a control word, the recognition is displayed and the recognition program signals *del* to prompt the user for the next input utterance.

The only one of the three control words which is essential for the operation of the system is "STOP". If this is not included in the vocabulary, there is no way for the user to terminate the recognition session (except by killing the process). The inclusion of "RETRAIN" in the vocabulary is necessary only if the option of replacing existing templates is desired. The benefits of including "CORRECTION" are that the output can be augmented (using the command facility described below) so that the preceding recognition is deleted when

"CORRECTION" is recognised; that it permits a form of supervised adaptation (as described in section 5.3); and that occurrences of "CORRECTION" in the results file can be used subsequently to identify misrecognitions (on the assumption that "CORRECTION" itself is recognised reliably).

This mode of operation (without verification by keyboard input) has the advantage that the user does not need to use the keyboard during the recognition session (unless retraining is required), and can thus perform a "hands-busy" task, or one which involves moving away from the keyboard, while using the recogniser. Because each recognition does not have to be verified explicitly before the next utterance can be taken in, the process of using the system is simpler and more convenient, as long as the input is generally recognised correctly. However, the correction procedure when a recognition is incorrect is slightly more time-consuming in this mode than with the verification option, because at least two additional utterances ("CORRECTION", and the repetition of the misrecognised word) must be recognised to accomplish a correction.

Perhaps the main disadvantage of the mode without explicit verification is that the control words may not be recognised reliably, and the consequences of this are particularly serious. If "STOP" cannot be recognised correctly, the user will not be able to end the recognition session (except by the unsatisfactory method of killing the process on the computer). If "CORRECTION" cannot be recognised, any misrecognition which may occur cannot be corrected. Also, if any other word is misrecognised as "STOP" or "CORRECTION", the session will be terminated early or a correct preceding recognition will be treated as incorrect. These problems can be overcome by replacing the templates which are causing the errors (after starting a new session in the case of early termination), using the retraining facility. (If the vocabulary includes words which are easily confused with the control words, it may be preferable when retraining to

substitute synonyms (e.g. "FINISH" and "WRONG") for the control words to improve the accuracy; these will be interpreted by the system as being simply new pronunciations of the original control words.) However, if "RETRAIN" cannot be recognised successfully, this retraining facility will not be available, and a separate training session to create new templates will be required before the recogniser can be used effectively.

Another disadvantage, though a less serious one, of the mode using spoken control words is that templates for these words must be included in the system: this adds slightly to the length of the training session, and also adds to the computation for each input utterance (since the input must be compared with all the templates), and increases the range of possible misrecognitions and hence the error rate.

This mode, as implemented, does not permit inspection of the second-best recognition candidate, or abandonment of the input utterance. These possibilities could in principle be incorporated by providing additional control words such as "NEXT" and "ABANDON" at the risk of making the system less simple to use.

Because of the potentially more serious consequences of misrecognitions when the system is being used without explicit verification, it is useful, for this mode of operation, to have a rejection capability, to allow a "no recognition" decision when the recognition is not reliable. This is accomplished by setting appropriate rejection thresholds, to be applied to the ratios of word distances, as described in section 4.3.2. When the system is being used with verification, the rejection thresholds may be set to 1.0 so that rejection does not occur; any wrong recognition can easily be eliminated by the user. When verification is not in use, the rejection threshold at the final comparison stage should be set to some value greater than 1.0. A rejection threshold slightly greater than 1.0 at the

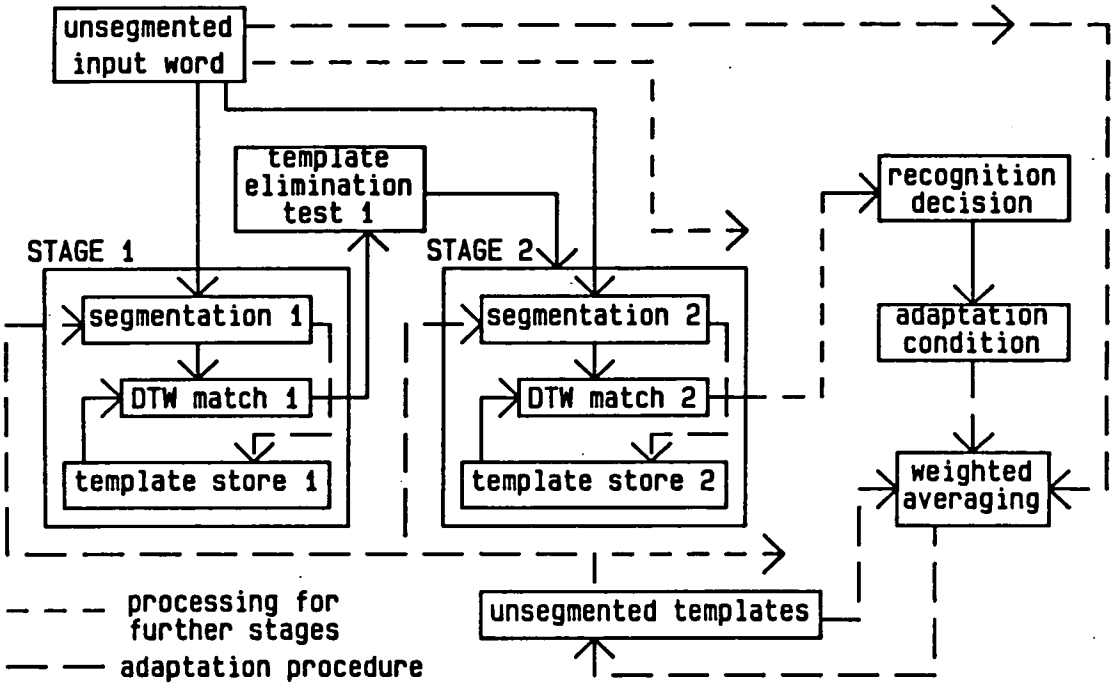
second stage may also be desirable, to allow early rejection of inputs whose recognition is very uncertain, and thus save computation time.

The system has a command execution facility built into it, to allow user-specified commands to be run in response to the spoken input. If this facility is in use, when any word is recognised, a corresponding Unix shell script is executed, using the "system" command within *awr*, before the recognition session continues. This command facility has been used mainly to enhance the output of the system during interactive recognition sessions: the shell command corresponding to each non-control word simply writes that word in a window on the terminal screen (separate from the window in which the direct output of *awr* appears), and the command corresponding to "CORRECTION" deletes the most recent word from that window, while the commands for "RETRAIN" and "STOP" do nothing. Thus the user can see the details of all recognitions (including those of control words), and of the retraining procedure when it is called, in the window used for the direct output of *awr*, but can also see just the recognised sequence of non-control words – with any corrections that have been made – in the auxiliary window. Although the application of the command facility in these experiments has been limited to this enhancement of the output, the mechanism implemented would allow the execution of much more extensive commands if desired.

5.3: Template adaptation

Options have been built into the recognition system which permit supervised or unsupervised adaptation of the templates, to incorporate information from the recognised input, during a recognition session. The structure of the adaptive word recognition system is shown in figure 5.2.

Figure 5.2: adaptive isolated word recognition system



When a word has been recognised, an adaptation condition is applied, and if this condition is satisfied the template corresponding to the recognition is adapted by weighted averaging with the input word. There are various types of adaptation condition, and the adaptation itself can take any of a variety of forms. These adaptation options are described in detail below.

5.3.1: Adaptation conditions for supervised and unsupervised adaptation

The forms which the adaptation condition can take depend on the verification available. If immediate verification of each recognition by the user is in operation, as in the first of the two interaction modes described in section 5.2, the condition can be imposed that the recognition must be correct. (That is, *supervised* adaptation can be implemented.) In this case also the template can be adapted negatively, to make it less like the wrongly recognised input, if the

recognition is incorrect. If the second-best candidate word is also found and verified when the best-matching template is incorrect, the template for this second-best candidate can be adapted too, positively or negatively depending on whether it is correct or incorrect. (There is also a modified option for negative adaptation, in which an incorrect first-candidate template is not adapted negatively unless the second candidate is correct. This is intended to prevent adaptation away from utterances which are badly affected by noise or endpoint detection failure.)

If no information from the user as to the correctness of the recognition is available, the adaptation condition must be based on some other source of information. The condition adopted in this case (for *unsupervised* adaptation) is that the ratio of the word distances for the second-best recognition candidate and for the best candidate must exceed a specified threshold. If the ratio exceeds this threshold, the recognition is assumed to be reliable, and the template for the best candidate word is adapted (positively) to the input. Otherwise, the recognition is assumed to be insufficiently reliable, and no adaptation takes place. (In fact, such a distance ratio criterion can also be imposed as an additional condition in the case where the correctness of the recognition is known; this might be useful to prevent adaptation to inputs which, though correctly recognised, were affected by noise or endpoint detection failure.)

In the mode of operation without immediate verification, a modified form of supervised adaptation can still be achieved, provided that the word "CORRECTION" is included in the vocabulary. In this case, when a word other than a control word is recognised, the input is not used immediately for adaptation, but is stored, along with its recognised identity, until the next recognition is obtained (corresponding to the next input utterance, or to some subsequent one if the next is rejected). If this next recognition is "CORRECTION", the

preceding recognition is assumed to be incorrect, and negative adaptation may be applied using the stored data. Otherwise, the preceding recognition is taken as correct, and the stored word is used to adapt (positively) the template corresponding to its recognised identity – unless the new input is recognised as "RETRAIN", in which case the stored information is discarded. Thus, provided there are no recognition errors involving the control words, the template corresponding to each recognition obtained can be adapted positively or negatively according to its correctness or incorrectness – with a one-utterance delay in the adaptation, to allow the user to say "CORRECTION" if the recognition was incorrect. The templates for control words, however, are adapted immediately on recognition of these words (subject to a distance ratio condition). This immediate adaptation is particularly necessary in the case of the word "STOP", since the recognition session is terminated when this is recognised and so there is no following utterance to be used for verification. (Supervised adaptation of "STOP" would require a slightly more complex session ending procedure, for instance with keyboard input requested to confirm or deny that the word "STOP" had been spoken.) But also, it was decided that the benefit of implementing delayed verification and adaptation for "CORRECTION" and "RETRAIN" would not justify the probable confusion that it would cause to the user. The harmful effect of mistaken adaptation of the template for "RETRAIN" can be overcome quite easily: if any word is misrecognised as "RETRAIN", then the retraining procedure automatically invoked as a result of this misrecognition can be used to replace the wrongly adapted "RETRAIN" template with a new one.

5.3.2: Selection of the template to be adapted

When the adaptation condition is satisfied, the template to be adapted is identified, and (subject to certain conditions on the lengths of the template and the input word) the weighted averaging operation is applied. This results in a new template which is stored for subsequent use in place of the old one.

If there is only one template for each word of the vocabulary, the selection of the template to be adapted is straightforward: it is simply the template for the specified word of the vocabulary (which is the best or second-best recognition candidate). In the case of the best candidate word, this is the template which has been found, during the recognition procedure, to match the input with the minimal distance (at the stage of comparison at which the recognition decision is reached); in the case of second-best candidate adaptation, it is the template with the second-smallest distance.

If there are two or more templates per word, however, then there are various possible criteria for selecting one of a given word's templates to be adapted. Two options have been implemented. In the first of these, the template adapted is the template, from the set of templates for the specified word of the vocabulary, yielding the minimal distance. (For first-candidate adaptation, this is, as in the case with only one template per word, the template with the minimal distance out of all the templates in use. For a second-best candidate word, however, it is no longer necessarily the template with the second-smallest distance, as this may belong to the first candidate word.) In the second option, the template adapted is the next template after the minimal-distance template (in the arbitrarily ordered template list) representing the same word of the vocabulary [238,239]. (If the minimal-distance template is the last in the list for the specified word, the first template in the list for that word is used.) This

"skewed" adaptation is intended for use in the case where the adaptation is unsupervised, to improve the stability of the system: if one of the templates for one word of the vocabulary (word A) is similar to the pronunciation of another word (B), so that an utterance of B is misrecognised as A, then the template for word A which is wrongly adapted to this input is not the template which was already similar to B and so caused the misrecognition, but another "A" template (which was less like the input utterance), and so the risk of recurrence of the same misrecognition is not increased as much as it would be by adaptation of the minimal-distance template.

5.3.3: The weighted averaging procedure

When the template to be adapted has been identified, a word length test is applied. No adaptation is performed if the lengths (in frames) of the template and the input are so different that alignment is impossible, or if the adaptation would result in an adapted template too long to fit into the designated region of the template data array. (The first of these conditions is necessary only if the DTW method of alignment is to be used in the adaptation: linear alignment can accommodate any disparity in word lengths.)

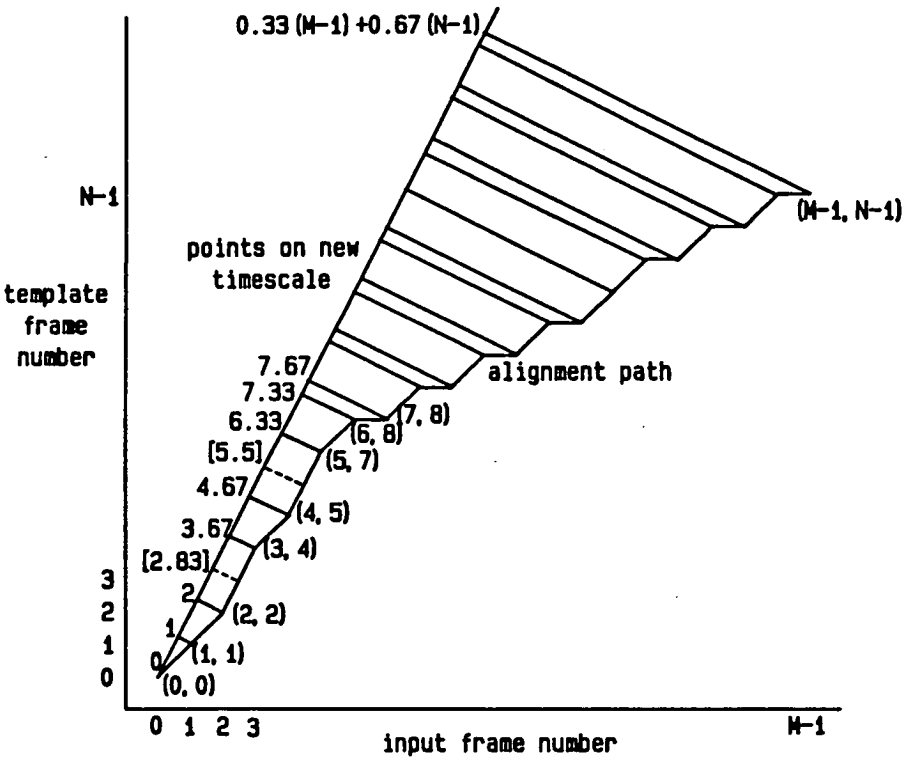
If the word length conditions are satisfied, the weighted averaging procedure is invoked to construct the new adapted template from the old template and the input. The parameters controlling the weighted averaging are the choice of alignment (DTW or linear) and the weights assigned to the existing template and to the input.

In the following description of the weighted averaging procedure, the convention is adopted that frame numbers in a template or an input word start from 0, rather than from 1. Thus the first frame of an N-frame template is

frame 0, and the last is frame $N-1$.

The details of the DTW algorithm used in the non-linear alignment option are the same as in the recognition phase (except that the alignment is not preceded by segmentation), and have been described in sections 4.2.2 and 4.3.2. During the DTW computation, a record is kept of the predecessor to each point on a possible alignment path, so that the optimal path can be recovered when the final point is reached. Once the alignment has been found, the input and template vectors matched together at each point on the path are averaged, with the weights which were given as parameters to the adaptation procedure. Also, at each step of the form (1,2) in the path, from $(m-1,n-2)$ to (m,n), an extra vector is interpolated between the input vectors for frames $m-1$ and m, and this is averaged with the vector for frame $n-1$ in the template. For each averaged vector, a time is computed on the timescale of the new template. The time (in frames) for the vector derived from input vector m and template vector n is $wm+(1-w)n$, where w and $1-w$ are the weights on the input word and the template respectively. This corresponds to projection of the point (m,n) in the input-reference plane onto a line of gradient $\frac{(1-w)}{w}$, as illustrated (for the case where $w=\frac{1}{3}$) in figure 5.3. In fact, the distance along this line between points t apart on the new timescale is $\frac{t}{(w^2+(1-w)^2)^{\frac{1}{2}}}$. The first averaged vector is taken as the first vector in the adapted template. (Its position on the new timescale is 0 unless any input vectors have been matched to the pseudotemplate frame (as described in section 4.3.2) at the beginning of the template.) Subsequent vectors are interpolated at intervals of 1.0 on the new timescale. If the position of the last averaged vector on the timescale exceeds the position of the last of these interpolated vectors by more than 0.5, this averaged vector is

Figure 5.3: weighted averaging operation for template adaptation



appended to the sequence of interpolated vectors. The resulting sequence of vectors constitutes the adapted template. The number of vectors in the adapted template is the nearest integer to $wM + (1-w)N$, where N is the number of vectors in the template before the adaptation and M is the number of input vectors which are matched to template vectors. (M is not necessarily equal to the number of vectors in the input, because of the possibility that some of the frames at the beginning and end of the input may be matched to pseudotemplate frames and thus excluded from the averaging. The extent to which this happens is controlled by the value of the constant distance assigned to pseudotemplate frame matching.)

During the experiments described in chapters 6 and 7, the version of the weighted averaging function in use contained a minor flaw, whereby the first

vector in the (unsegmented) adapted template was replaced by either a zero vector or a vector left in the array from the previous invocation of the function. This also affected the averaging of training utterances to form initial templates. It was discovered too late to allow the experiments to be rerun. However, a comparison of the uncorrected and corrected versions was made, using the digits spoken by the 49 test speakers, with template set D6 (as in chapter 7), and the difference was not found to be significant. The correction improved the recognition result with supervised adaptation (over four random orders of the input for each test speaker) from 94.69% to 94.71%, but reduced the recognition accuracy with unsupervised adaptation from 93.37% to 93.33%. (Neither of these accuracy differences was statistically significant.)

If linear alignment instead of DTW alignment is specified for the adaptation procedure, one point (m,n) is defined in the alignment path for each input frame number m; the value of n is chosen so that $\frac{n}{m}$ is as close as possible to $\frac{N-1}{M-1}$, where N and M are the numbers of vectors in the template and the input respectively. (With this linear alignment there is no provision for omission of frames at the beginning and end of the input.) The weighted averaging then proceeds as in the case with DTW alignment. If the word length ratio $\frac{N-1}{M-1}$ exceeds 2.0, steps of the form $(1,k)$ will occur, where $k > 2$. In such a case, where successive points on the path are $(m-1,n-k)$ and (m,n), the procedure adopted is similar to that for a step of the form (1,2): $k-1$ extra vectors are interpolated, equally spaced between input vectors $m-1$ and m, and are averaged with vectors $n-k+1$ to $n-1$ from the template. The same procedure as in the DTW case is employed to derive vectors on a new timescale, which form the adapted template.

Once an adapted template has been formed by this weighted averaging process, it is stored in place of the previously existing template from which it was obtained. Segmented versions of the adapted template are derived, and are stored, in place of the segmented versions of the unadapted template, for use in the recognition of subsequent input utterances (as shown in figure 5.2).

5.3.4: Adaptation weighting options

The weights (w and $1-w$) on the input data and on the existing template control the extent to which the template is adapted to the input utterance: the greater the value of w, the greater the amount of adaptation. In the special case where $w = 1.0$, the adaptation becomes simply a replacement of the template by the input. If $w = 0$, the adapted template is the same as before adaptation. For negative adaptation, a negative value of w is adopted – typically about -0.05 – and so the weight $1-w$ on the existing template is greater than 1. Apart from the fact that w is negative, the alignment, averaging and timescale definition procedure is exactly the same for negative adaptation as for positive adaptation.

Two options are provided for the weighting of the template and the input at successive adaptations. The two forms of weighting are referred to as "tracking" and "optimisation".

In the tracking formulation, the weight w on the input is kept constant ($w = w_0$) for all (positive) adaptations of any template. Thus, after the first adaptation of a particular template, the adapted template is a weighted average of the original template and the input used to adapt it, with weights $1-w_0$ and w_0 respectively. After a second adaptation (to a subsequent input), the template is a weighted average of the previous adapted template and the new input, with weights $1-w_0$ and w_0 again – or, equivalently, a weighted average of the origi-

nal template and the two inputs, with respective weights $(1-w_0)^2$, $w_0(1-w_0)$ and w_0. At each subsequent adaptation, the weight of the original template's contribution to the adapted template is reduced by a factor of $1-w_0$, as is the weight of each earlier adaptation input's contribution, while the most recent input is given weight w_0. Thus the contribution of each utterance (whether from the initial training session or from the adaptive recognition input) to the template in use decays exponentially with successive adaptations. The rate of this exponential decay (relative to the rate of occurrence of adaptations of a particular template) is $-\log(1-w_0)$: it is more rapid, the larger the value of w_0.

In adaptive recognition with this tracking form of weighting, the current form of each template depends most on the most recent input utterances to which it has been adapted. This form of adaptation is suitable for tracking gradual variations in the acoustic realisation of a word. Such variations may occur during a protracted recognition session, owing to the effects of fatigue on the user, or perhaps because of gradual changes in the level and characteristics of background noise, or in other aspects of the acoustic conditions such as the distance from the user's mouth to the microphone. On a longer timescale, if the same person is using a recognition system on different occasions, there is likely to be some drift in the user's voice characteristics and habitual pronunciations of words. The value of w_0 to be chosen for use in any particular application will depend on the typical timescale of the variations which the adaptation is intended to follow, and also on the rate at which adaptations of a template occur in time. (The greater the number of templates in the system, the less frequently, on average, each one can be adapted during a recognition session.) A large value of w_0 allows rapid adjustment to changes in voice, pronunciation or acoustic environment, but may also make the system vulnerable to corruption of templates through adaptation to atypical, noisy, inaccurately endpoint-detected

or (in the case of unsupervised adaptation) misrecognised inputs. Thus the optimal value of w_0 will depend on the tradeoff between the speed of adaptation desirable and the requirement for stability of the templates.

In the optimisation formulation, the weights on the input and the template are adjusted at successive adaptations, so that the relative weight on the template (i.e. the ratio of the template weight to the input weight) increases according to the number of utterances which have been used so far to form it. (For each template in the system, a record must be kept of the number of positive adaptations performed so far. No account is taken of negative adaptations.) If, in the first (positive) adaptation of a particular template, the relative weight on the unadapted template is v_0, then, when the template is adapted (positively) for the nth time, the weights are adjusted so that the relative weight on the template (which has already been adapted n-1 times) is

$$v_{n-1} = v_0 + (n-1). \tag{5.2}$$

After n adaptations, the template is a weighted average of the original template and the n input utterances used to adapt it, in which the original template's contribution has weight $\frac{v_0}{v_0+n}$ and the contribution from each of the n input utterances has weight $\frac{1}{v_0+n}$. In particular, if v_0 is set to 0 (so that the original template is replaced by the recognised input in the first adaptation), then the adapted template (after any number of adaptations) is simply the average of the input utterances used in forming it. If v_0 is set to 1.0, then the adapted template is the average of the original template and all the inputs used to adapt it.

The weight w_n on the input in the nth adaptation is related to the relative weight on the template, v_{n-1}, by the equation

$$w_n = \frac{1}{v_{n-1}+1}; \tag{5.3}$$

the weight on the template is

$$1-w_n = \frac{v_{n-1}}{v_{n-1}+1}. \tag{5.4}$$

This is because, when the input weight is w, the template weight is $1-w$, and hence the relative weight on the template, v, is given by

$$v = \frac{1-w}{w} = \frac{1}{w} - 1. \tag{5.5}$$

The two special cases mentioned above, where v_0 is 0 or 1.0, correspond to setting w_1 to 1.0 or 0.5, respectively. If a value of w_1 smaller than 0.5 is adopted, to improve the stability of the template at its first adaptation, then the resulting value of v_0 will be greater than 1.0. Whatever the value of v_0, however, the template after a sufficiently large number of adaptations will closely resemble an average of all the inputs used to adapt it. (This statement must be qualified slightly, in that the ultimate length of the template after many adaptations can be substantially affected by the value of v_0; this is discussed in detail below.) Thus the optimisation form of weighting is appropriate for adaptation whose goal is not to track gradual changes in the typical realisation of a word, but to optimise the conformity of the template to a typical (or average) realisation of the word which is assumed to be invariant with time.

The length, in frames, of the adapted template becomes fixed after the nth adaptation, for the first value of n for which

$$w_{n+1} < \frac{0.5}{\max\{L_n - L_{\min}, L_{\max} - L_n\}}, \tag{5.6}$$

where L_n is the length of the template after n adaptations, and $L_{\min}$ and $L_{\max}$ are the minimum and maximum possible lengths of input words to be used in subsequent adaptations (which depend on the parameters for digitisation and endpoint detection, and also on L_n if DTW alignment is in use) – because after this value of n the weighted-average length of template and input in any subsequent adaptation is always within 0.5 of the existing template length L_n. Thus the ultimate length of the template is determined by the lengths (and the order) of the initial template and a limited number of utterances at the beginning of the adaptive recognition. The value of n at which (5.6) is satisfied depends on the value of v_0; in particular, for large enough values of v_0, (5.6) is satisfied when $n = 0$, and so the length of the template is not altered at all by any adaptation. However, the actual values of the vectors making up the adapted template are still affected by all the input utterances used to adapt it, and so, disregarding the length normalisation, the template after a large number of adaptations is nearly the average of all the adaptation inputs. Another effect of word lengths, which occurs only when DTW alignment is used in the adaptation procedure, is that the length of the current template restricts the lengths of words which may be aligned to it, so that adaptation is not performed for inputs which are too long or too short. This restriction on the inputs which may be used for adaptation of a given template applies equally whatever the current adaptation weights are; but it is likely to be more severe in its effects if the value of v_0 is large, because then the length of any template which is (initially) particularly long or short will have less opportunity to be moderated by adaptation.

This discussion of word length effects has assumed the optimisation form of weighting. However, similar effects can occur with the tracking formulation. If w_0 is small enough to allow (5.6) to be satisfied for some value of L_n, the length of a template may become permanently fixed (after any number of adaptations from 0 upward); and in general, the smaller w_0 is, the more limited and less frequent the modification of template lengths will be.

In the supervised adaptation option where negative adaptation is applied in cases of misrecognition, the (negative) input weight used in this adaptation must be defined. In the tracking formulation, this weight, like the input weight for positive adaptation, is read in from the parameter file at the beginning of the recognition session and is not changed thereafter. In the optimisation formulation, the weight for negative adaptation is adjusted so that the relative weight on the template in negative adaptation increases linearly, as the template is adapted positively, with the relative weight on the template for positive adaptation.

5.3.5: Compensation for adaptation

It was found in experiments with adaptive recognition that, as any template was adapted, the distances obtained in comparing this template with input utterances tended to become smaller, not only in the case of input words for which the template represented a correct recognition, but also in the case of other input words. This effect of adaptation can lead to recognition errors, where the template for the correct recognition of a particular input has not yet been adapted (or, with multiple templates, where none of the templates for the correct recognition has been adapted) but some incorrect template has been: the adapted incorrect-candidate template may be closer to the input than any correct

template, so that the input is recognised as the (incorrect) word whose template has been adapted. (Details of these experiments and their results will be given in sections 6.3.2.1 and 7.3.1.) To compensate for this effect on the recognition accuracy during adaptive recognition, provision was introduced into the system for the adjustment of each word distance obtained (at any of the comparison stages) according to the amount of adaptation previously applied to the template being matched. When any template is matched with an input utterance, at any of the stages of the recognition process, the distance obtained is multiplied by a quantity (a compensation factor) which depends on the number of times that template has been (positively) adapted. The compensation factors can be specified for values of n (where n is the count of adaptations of the template) from 1 to 19. (The compensation factor for $n = 0$ is fixed at 1.0.) If the maximum value of n for which a compensation factor is specified is n_{max}, then the compensation factors for larger values of n are made the same as the factor for n_{max}. The compensation factors are chosen so that they increase with n.

5.3.6: Word distance normalisation

Another form of adaptation – not template adaptation, but adaptive word distance normalisation – was introduced into the system to take account of the possible differences in typical correct-template distances for different words of the vocabulary. A normalisation quotient is defined for each template, for each comparison stage. Initially, the quotients for all templates (for any particular comparison stage) are the same. When an input is recognised, and (if verification is in operation) the recognition is verified as correct by the user, the quotient for the best-matching template, at each comparison stage used in this recognition, is adapted towards the distance value (after any compensation)

obtained in matching the template at that stage. The default quotients, for templates which have not yet been selected as correct recognitions of any input, are also adapted, so that for each comparison stage the default quotient is the average of the quotients which have been adapted. Then, for each new input utterance, the distances obtained from the comparison (at any stage) are normalised, by dividing by the current values of the appropriate quotients, before being used to determine the recognition of the input. The intention of this procedure is that the normalisation should compensate for any bias, in the recognition decision, in favour of templates yielding smaller distances.

If a template is retrained during a recognition session, any counts of recognitions or adaptations associated with that template (for the purpose of determining normalisation quotients, adaptation weighting or compensation factors) are reset to 0.

5.4: Experiments with recognition system parameters

The word recognition session described in the previous sections of this chapter incorporates a number of components which have parameters to be specified. Some of the parameters define the form and rate of template adaptation, and the amount of word distance compensation required for any particular case of adaptation; as template adaptation is the main topic of the research reported in the next two chapters, the choice of values for these parameters will not be discussed in this section. Values for some of the other parameters have already been specified, in sections 4.2.4, 4.4.4 and 5.2. However, there remain certain parameters whose values must be defined, both in the acoustic representation of each frame of speech and in the recognition processing. This section describes briefly some experiments conducted to determine appropriate settings

of these parameters, and states the parameter values adopted for the work described in the subsequent chapters.

In the LPC analysis component of the system, the order of the analysis and the number of cepstral coefficients output have to be specified. The higher the analysis order is, the more features of the spectrum can be modelled for each frame of speech; but also, as the order of analysis is increased, the amount of computation required in the acoustic processing expands. The number of cepstral coefficients output from the acoustic analysis determines the amount of detail in the linear predictive spectra which is available for use in obtaining frame distances, and hence word distances and recognition decisions; but the computation required in the recognition and adaptation processes increases with the number of coefficients per frame, since the computation for each frame distance, interpolation or vector averaging operation is approximately proportional to the vector length. Also, the storage requirements for analysed data increase with the number of coefficients per frame. Therefore, it is of interest to determine what analysis order must be used, and how many cepstral coefficients per frame are required, to obtain satisfactory recognition results.

Several combinations of analysis order and number of coefficients were compared. The analysis orders used were 8, 12 and 24; the number of coefficients per frame ranged from 8 to 24. Each order (p) of LPC analysis in turn was applied to the same set of sampled data, obtained from isolated utterances of words from two vocabularies (the 10 digits, and the 50-word vocabulary of numbers, days and months listed in table 4.4, with the control words "STOP", "RETRAIN" and "CORRECTION" added in each case), and p cepstral coefficients per frame were output. The representations of the data containing smaller numbers (c) of coefficients per frame were subsequently derived by selection of the first c coefficients per frame from each analysed data file. The

sampled data were collected during interactive recognition sessions, using an option in *del* which permitted output of each detected word to a sampled data file between the endpoint detection and LPC analysis stages. Similar processing was applied to training utterances to obtain appropriately analysed and represented versions of a number of sets of templates. Each combination (p,c) of analysis order and number of coefficients was then evaluated by recognition experiments, with and without template adaptation.

The results of these experiments with the analysis order and the number of coefficients per frame are shown in table 5.2. "F" and "W" are the codes for the two vocabularies (of 13 and 53 words respectively). All the training and test data were from a single speaker, who was the same as speaker 1 in the data bases described in section 4.4.2. Results are tabulated for recognition of "F" data collected during a single session, using four template sets (F1-F4), and for recognition of two sets of "W" data from separate sessions using two template sets (W1 and W2); the average recognition accuracies over the template sets for each vocabulary are also given. (All the template sets consisted of single-token templates, except F3, which was made up of two-token averaged templates.) The results identified by "0" in the "adaptation" column are with no template adaptation; those identified by "1" are with supervised adaptation, with the tracking formulation, input weight $w_0 = 0.2$, negative adaptation weight -0.05, and second-best template adaptation where the best-matching template is incorrect, but with no compensation.

The best recognition results overall were obtained with the 24th-order LPC analysis. On the "F" vocabulary, the accuracies attained with 24th-order analysis were generally better than those with 12th-order analysis. (When 12 coefficients were used, the average differences in recognition accuracy between

Table 5.2: recognition accuracies with different analysis orders and numbers of cepstral coefficients per frame

Input (number of words)	Template set and adaptation	LPC order and number of cepstral coefficients (p,c)								
		(8,8)	(12,8)	(12,10)	(12,12)	(24,8)	(24,10)	(24,12)	(24,16)	(24,24)
F (119)	F1 0	86.6	87.4		88.2	86.6	90.8	92.4		92.4
	F1 1	88.2	89.1		90.8	89.1	91.6	93.3		94.1
	F2 0	85.7	87.4		89.9	88.2	90.8	91.6		93.3
	F2 1	89.1	89.9		93.3	92.4	93.3	95.8		94.1
	F3 0	90.8	94.1		96.6	96.6	95.8	97.5		97.5
	F3 1	92.4	92.4		94.1	94.1	95.0	94.1		95.8
	F4 0	86.6	86.6		89.9	85.7	87.4	89.1		94.1
	F4 1	89.9	89.9		92.4	90.8	93.3	95.0		95.0
	ave 0	87.4	88.9		91.2	89.3	91.2	92.6		94.3
	ave 1	89.9	90.3		92.6	91.6	93.3	94.6		94.8
W (a) (224)	W1 0	79.5	80.8	79.9	79.5	81.7	80.8	81.2	81.2	
	W1 1	83.9	84.4	83.5	83.5	84.4	82.6	83.5	84.4	
	W2 0	88.4	89.7	90.6	91.5	89.3	92.0	92.0	90.6	
	W2 1	86.6	88.8	91.5	90.6	90.2	91.5	90.6	90.2	
	ave 0	83.9	85.3	85.3	85.5	85.5	86.4	86.6	85.9	
	ave 1	85.3	86.6	87.5	87.0	87.3	87.1	87.0	87.3	
W (b) (89)	W1 0	82.0	80.9	84.3	84.3	82.0	84.3	82.0		
	W2 0	74.2	77.5	77.5	77.5	76.4	76.4	76.4		
W (a,b) (313)	ave 0	82.3	83.6	84.0	84.2	83.7	84.7	84.5		

the cases of 12th-order and 24th-order analysis were 1.5% (without adaptation) and 1.9% (with adaptation); the standard errors of these figures, estimated from the variation across template sets, were 1.04 and 0.63 respectively, yielding confidences 0.88 and 0.97.) On the "W" vocabulary, the results attained with 24th-order LPC did not differ significantly from those with 12th-order LPC. On

both vocabularies, 8th-order analysis gave poorer results than 12th-order or 24th-order even when only the first 8 coefficients from the higher-order analysis were retained.

For a given analysis order (12 or 24), the accuracy was improved in most cases by an increase in the number of cepstral coefficients retained for use in the recognition processing. For the "F" vocabulary, the results with 10 or 12 coefficients were nearly always better than the corresponding results with 8 coefficients, and (with 24th-order analysis) there was a further improvement in accuracy (along with a larger increase in the computational and memory requirements) with retention of the full set of 24 coefficients. For the "W" vocabulary, the results were less consistent, and there was little difference on average among the results with differing numbers of coefficients.

A side-effect of the use of different cepstral representations in these experiments was that the numbers of templates retained for matching at the second and third stages in the recognition procedure varied according to the number of coefficients per frame and the order of the LPC analysis by which they were obtained. In most cases, as the number of coefficients per frame was increased, the word distance ratios determining the elimination of templates became closer to 1.0, and in consequence the numbers of templates retained at the later stages of comparison were increased. Given the design of the three-stage recognition system, it is difficult to assess how much of the improvement observed with an increase in the number of coefficients was due to this retention of more templates for detailed comparison. (This effect could be eliminated by using only one comparison stage, but this would make the results less directly applicable to the three-stage system. Ideally, the acoustic representation parameters and the segmentations and thresholds in the multiple-stage decision procedure should be optimised together, rather than sequentially, but this might be difficult in

practice because of the large number of parameters to be combined.)

A full assessment of the effects of the analysis order and the number of cepstral coefficients retained would have required more extensive experiments with a larger data base (preferably collected from several male and female speakers). In particular, it might be of interest to explore the effects of analysis orders between 12 and 24, and to determine whether the differences between the "F" and "W" vocabularies' results were due to characteristics of the vocabularies, or merely to peculiarities of the specific sets of data collected for the experiments. However, as the optimisation of the acoustic analysis and frame representation parameters was not the main goal in view, but was merely a preliminary step to the study of template adaptation, no further experiments were conducted to explore these topics. For subsequent work, the LPC order was fixed at 24, and the number of cepstral coefficients output for use in recognition was fixed at 12. The choice of 24th-order analysis is in agreement (given the 20kHz sampling of the input speech) with the recommendation, on theoretical grounds, that the analysis order should be at least $n+4$ when the sampling rate in kHz is n [20:p.154]; it was also the same analysis order which had been adopted for other work within the Centre for Speech Technology Research. The decision to use 12 cepstral coefficients per frame represented a compromise between optimisation of the recognition accuracy (which would have demanded a larger number of coefficients) and a compact representation which would improve the speed of vector distance computations.

Some experiments were conducted with weighting of the cepstral coefficients, as this had been found by other researchers [39,40] to improve recognition accuracies. The weighting was applied to the stored coefficients before they were used as input to the recognition program. (Thus the weighting operation had to be applied only once for each frame of speech, rather than

every time a distance was computed, but the effect was the same as that of using a weighted distance measure with fixed weights.) Two types of weighting were tested: weighting of each cepstral coefficient by the reciprocal of its standard deviation, found over a large number of frames of speech [39], and weighting of the ith coefficient (for each i from 1 to 12) by a quantity of the form

$$w_i = 6\sin\frac{\pi i}{12} + 1 \qquad (5.7)$$

[40]. The reciprocal-standard-deviation weighting, applied to 500 words from the 50-word vocabulary (table 4.4) and one set of templates, with each frame represented by 8 cepstral coefficients, and with the standard deviations estimated from the same data used as test input, reduced the recognition accuracy from 90.0% to 85.8%. The weighting defined by (5.7), applied to 300 words from the same vocabulary and to each of three template sets, with 12 cepstral coefficients per frame, reduced the average accuracy from 89.0% to 87.6% – though there were also reductions in the numbers of templates matched to the input at the second and third stages of the recognition procedure. Further experiments, with adaptation, and with the thresholds in the recognition procedure adjusted to yield similar comparison statistics to those without weighting, still showed losses in average accuracy. Therefore no weighting of the cepstral coefficients was adopted for subsequent recognition experiments.

The adaptive word distance normalisation technique described at the end of section 5.3 was evaluated on utterances from the "F" vocabulary and the 50-word vocabulary. The results obtained were inconclusive; on average the recognition accuracies were very similar with and without the normalisation technique. No further work was done on word-specific distance normalisation, and the technique was not used in the subsequent experiments. It remains possible,

however, that the technique might yield some enhancement of recognition accuracy if tested over a larger number of template sets with long sequences of input utterances. (Improvements in recognition have been recorded [98] with a more sophisticated word-specific distance normalisation technique, which involves the use of a large set of training data to estimate parameters of the distribution of correct-match distances for each word of the vocabulary.)

CHAPTER 6

ADAPTATION OF SPEAKER-SPECIFIC TEMPLATES

6: ADAPTATION OF SPEAKER-SPECIFIC TEMPLATES

6.1: Introduction

Applications for an adaptive isolated word recognition system can be divided into two main classes: those where the initial templates are provided by the intended user (during a training session before the recognition session begins), and those where the initial templates are speaker-independent (so that no training session is required for a new user). In the first of these cases, the role of the adaptation during the recognition session is to improve or update templates which are already speaker-specific; in the second case, it is to make initially speaker-independent templates specific to the current speaker. The first of these two cases is considered in the present chapter, and the second in chapter 7.

(It is also possible to implement a system in which speaker-specific templates are formed for only some words of the vocabulary, and speaker-independent templates are used for the other words. Such selective training to a new speaker may be appropriate if there are some words in the vocabulary which are more frequent in the input, more variable from speaker to speaker, or more confusable than the other words, or some words for which reliable recognition is particularly important (such as the control words "STOP", "RETRAIN" and "CORRECTION" for instance). This intermediate case between speaker-trained and initially speaker-independent recognition is not explored here.)

The experiments reported in this chapter, with speaker-specific initial templates, were conducted using the adaptive recognition system described in chapter 5. The system was used firstly in its interactive mode for recognition sessions in which input utterances were collected to form a data base for subse-

quent experiments. The main series of experiments were conducted using the recognition system in its non-interactive mode, with the stored data as input. This use of stored data allowed comparison of different sets of adaptation parameters on the same input utterance sequences, and re-use of the same data in different orders and with different template sets to improve the statistical reliability of the results obtained. (A comparison of different adaptation parameters on the basis of recognition performance obtained during interactive recognition sessions would have required an impracticably large number of interactive sessions, to smooth out the random variations associated with particular utterances. However, recognition statistics for the interactive sessions (during which adaptation was applied) are given in section 6.5.)

In section 6.2, the interactive procedures for template formation and data collection are described, and the sets of utterances forming the data base are listed.

The experiments conducted using this data base are described in section 6.3, and the results obtained are presented and discussed in detail. Section 6.4 contains a more general discussion of the main findings of the experiments, and of options in adaptation of speaker-specific templates.

Section 6.5 contains some statistics on the interactive recognition sessions, and comments on the forms of interaction occurring between the users and the system.

6.2: Interactive training and data collection

6.2.1: Training procedure

For each speaker and vocabulary, the first requirement was the formation of a set of templates for use in the interactive recognition sessions. In fact, at least two template sets were formed, on separate occasions, for each speaker and vocabulary – one consisting of a single-utterance template for each word in the vocabulary, and one consisting of one template per word derived by a robust averaging procedure [30,31,255] from two utterances. The training procedure for forming a set of templates is as follows.

The speaker sits at a terminal, wearing a Sennheiser HME1019 headset which incorporates a microphone on an adjustable mounting. The microphone is positioned close to the mouth, but not directly in front of it (so as not to pick up breath noise). The volume setting on the recording equipment is adjusted so that the typical maximum amplitude of a spoken word is within the range of the analogue-to-digital convertor. (The amplitude of the signal can be monitored visually using an oscilloscope which is connected into the data path after the lowpass filter.)

The parameters required by the training procedure are the vocabulary, the number of utterances to be averaged to form each template and (in the case of more than one utterance per template) the threshold on distance per frame for utterance averaging. Once these have been specified, the training program displays one word at a time on the terminal screen and the speaker is prompted to utter the word. The training program operates in conjunction with *del*, in the same way that *awr* does, as described in section 5.2. If each template is to be formed from more than one utterance, the system prompts the speaker for more utterances of the same word, until two are obtained whose average distance per

frame (computed by DTW alignment) is below the threshold, and then averages the cepstral representations of the two utterances; if the number of utterances to be used per template is greater than two, further utterances are collected and averaged into the template as required. (This utterance collection and averaging procedure is very similar to that incorporated in the retraining facility within *awr*, as described in section 5.2.) Once the template has been formed, the training program displays the next word of the vocabulary, and the process is repeated. When templates have been formed for all the words of the vocabulary, the complete set of templates (consisting of time sequences of cepstral vectors) is stored in a file on the computer.

(Although an option exists in *del* to allow storage of the sampled data in a file between the endpoint detection and LPC analysis stages (as mentioned in section 5.4), this option was not used in most of the training and recognition sessions during the collection of the data base described here – partly because the writing of sampled data to disc memory slows the operation of *del*, making the session rather tedious for the speaker, and partly because of the large amount of disc space (40000 bytes per second, with 20kHz sampling) required for storage of sampled speech. The decision to store only the cepstral coefficients has the disadvantage that the data base cannot be played back – unless a suitable resynthesis program is available – for aural checking.)

The distance threshold for averaging was set, for each speaker, so that the number of extra repetitions required (because of excessive distances between the first two utterances) was fairly small. For two-token average templates, the average number of training utterances required per template ranged from 2.2 to 2.5 across different speakers and template sets.

The training sessions were conducted in a room containing several computer terminals, and on some occasions there were other people present, working at the terminals, besides the speaker and the experimenter. There was continuous background noise from the air conditioning equipment in the adjoining computer room, and there were occasional louder noises such as the opening and closing of doors. On one occasion the slamming of a door was detected as a word by *del* and a template was formed from it; this template was replaced, using the retraining option in *awr*, at the beginning of the first recognition session using that template set.

6.2.2: Interactive recognition sessions

For each speaker and vocabulary, once templates had been formed, several recognition sessions were conducted, on different days and at varying times of day. The format adopted for most of the recognition sessions (in which the vocabulary consisted of the 10 digits and the control words "STOP", "RETRAIN" and "CORRECTION") was that of a "data entry simulation", in which a list of digits was displayed on the terminal screen and the speaker had to reproduce that list by speech input. The interactive recognition system (as described in chapter 5), running in one window on a graphics terminal screen, was interfaced to another window on the same terminal (using the command facility described at the end of section 5.2) so that recognised digits were displayed in the second window and the most recently displayed digit was deleted on recognition of "CORRECTION". Any indication of rejected input, and any prompting for keyboard input in the retraining procedure, appeared only in the first window. Once the complete list of digits had been entered, the speaker terminated the session by saying "STOP".

In some of the recognition sessions for one speaker (speaker 1), a slightly different format was adopted, without the interface to the second window; but the task was the same – to enter a prespecified sequence of words, using "CORRECTION" in cases of misrecognition. In these sessions, the sequence of words was read from a sheet of paper, or in some cases from an additional window on the terminal which allowed scrolling of the text so that each word (or group of a few words) was revealed just before it was to be spoken.

In most cases, the same template set was used for consecutive sessions, for a given speaker and vocabulary. Template adaptation was applied during most of the recognition sessions; at the end of each adaptive recognition session, the adapted templates were stored, and these adapted templates were used in the next recognition session.

Most of the recognition and adaptation parameters were kept constant across different sessions. The parameters of the three comparison stages were as specified in section 4.4.4 (with the endpoint adjustment technique incorporated into the third stage in a few of the earlier sessions); the template elimination thresholds t_1 and t_2 were set at 1.6 and 1.2 respectively; and the distance ratio thresholds r_2 and r_3 to ensure rejection of unreliably recognised input were set at 1.05 and 1.15. (Lower thresholds were tried, but the higher values were restored after two recognition sessions because of the danger of errors: the lower value of r_3 had allowed an utterance of "6", affected by noise, to be recognised as "STOP".) The delayed adaptation option was adopted, to allow supervised adaptation without explicit verification of each recognition (as described in section 5.3.1). The adaptation used the tracking form of weighting, with w_0 equal to 0.2, and incorporated DTW alignment. No negative adaptation was employed. Various compensation factors were used; in particular, smaller compensation factors were applied when the templates in use had already been adapted during a

previous session than when they were initially unadapted.

The environment and conditions for the interactive recognition sessions were the same as for the training sessions. Occasions when particular utterances were badly affected by noise (and the words were therefore rejected or misrecognised, and had to be repeated) were noted, to allow exclusion of these utterances from the data files for subsequent experiments.

6.2.3: Vocabularies and speakers

Two vocabularies were adopted for the data collection: the digits ("0" to "9" – with "0" pronounced "zero") with the control words "STOP", "RETRAIN" and "CORRECTION" (a 13-word vocabulary, denoted by the vocabulary code "F"); and a 53-word vocabulary (denoted by "W") consisting of the numbers, days and months (as listed in table 4.4) and the three control words. The corresponding vocabularies without the control words are denoted by "d" (for the 10 digits) and "t" (for the 50 numbers, days and months). The vocabularies including the control words ("F" and "W") were used in the interactive training and recognition sessions, and then utterances of the non-control words (belonging to vocabularies "d" and "t" respectively) were extracted from the stored data obtained in these sessions to make up the data base for subsequent experiments.

Four speakers contributed to the data base: two male (speakers 1 and 2) and two female (speakers 3 and 4). Speaker 1 was the same as speaker 1 in the data base for the segmentation experiments described in section 4.2. All the speakers were members of the Centre for Speech Technology Research, and had had some experience of using the isolated word recognition system before the collection of this data base; they represented a variety of British accents (two being Scottish – speakers 1 and 4 – and the other two English).

6.2.4: Details of data base

The digits portion of the data base consists of utterances by each of the four speakers. The "t" vocabulary portion contains utterances by speaker 1 only.

The digits data base includes two sets of templates for each speaker, obtained by the interactive training procedure as specified in section 6.2.1 – one consisting of a single utterance of each word, and the other consisting of two-token averaged templates (one per word). It also includes 500 test utterances (50 repetitions of each of the 10 digits) by each of the four speakers. For each speaker, the sequence of 500 test utterances consists of 10 repetitions of a standard 50-digit sequence, devised so that no sequence of two digits occurs more than once in it. (The sequence is "2 0 4 1 5 7 3 9 8 6 5 3 2 1 7 9 4 0 7 6 1 9 5 5 4 9 2 7 8 0 3 3 6 7 2 4 5 0 8 9 1 0 6 6 3 4 8 1 2 8".)

In each interactive recognition session using the "data entry simulation" format, the standard sequence of 50 digits was displayed as the set of data to be entered. Thus 10 sessions were required for each speaker; in each session, the input consisted of the specified sequence of 50 digits, with repetitions and utterances of "CORRECTION" and "RETRAIN" where necessary, and an utterance of "STOP" to conclude the session. In fact, for speaker 1, not all the sessions had this form: the first five repetitions of the 50-digit sequence were collected in six sessions (without the two-window data entry simulation), four of which (all on the same day) each included only part of the sequence (the four partial sequences composing two complete repetitions), and one of which included two repetitions of the sequence. For speaker 2, also, one of the repetitions was in two sessions (though with only a brief pause between them), because the first session ended early through the recognition of "6" as "STOP" (mentioned in section 6.2.2 above). Apart from these cases, however, each session provided one

complete repetition of the 50-digit sequence, and all sessions for a given speaker were on different days. Details of the individual recognition sessions are presented in section 6.5.

Where a session included repetitions of a word in the sequence because the first utterance was rejected or misrecognised, the first utterance of the word was selected for inclusion in the data base, unless it had been noted as being badly affected by noise or endpoint detection failure, in which case the next utterance of the word (not marked as noisy or badly detected) was selected instead. In the whole 2000-digit data base, four first utterances were excluded as noisy or badly detected.

(The digits data used for the multiple-stage recognition experiments already described (section 4.4) consisted of the first six repetitions of the 50-digit sequence from each of the four speakers – except for one of the 50-digit subsequences for speaker 1: in this case, the 50 utterances used in the multiple-stage experiments, which came from three sessions (because of repeated recognition of "7" as "STOP"), were replaced by another set of 50 utterances when the larger data base was constructed.)

The "t" vocabulary portion of the data base consists of various sequences of words provided by speaker 1 in interactive recognition sessions with the "W" vocabulary. Some of the sessions used a standard ordering of the 50 non-control words, repeated a varying number of times per session; in others, a longer target input sequence was generated randomly or to simulate a possible data entry task. Several input sequences for subsequent experiments were constructed from the utterances collected in these sessions. In particular, six repetitions of the standard 50-word sequence, obtained from six recognition sessions over a period of nearly two weeks, were concatenated to form the 300-word sequence used in the experiments of section 4.4. (This sequence was also used for some

further experiments, and is designated "t4" in section 6.3.2.3.) The rule for selecting utterances from the input in cases of repetition was the same as in the case of the digits, as stated above. In the construction of the 300-word sequence t4 and of the 1000-word sequence t3 (used in the experiments of section 6.3.2.1), the number of first utterances excluded by this rule was eight in each case. Details of all the "t" vocabulary sessions, and of additional digit recognition sessions with randomly-ordered sequences spoken by speaker 1, are given in section 6.5.

6.3: Experiments and results

6.3.1: Design of experiments

In experiments with template adaptation, unlike most other forms of isolated word recognition experiments, the order in which the input utterances are presented to the recognition system is significant. Details of the input sequence ordering can affect both the recognition accuracy obtained during the adaptive recognition and the adaptation of the templates. The recognition of any given utterance depends on the current state of the templates, and therefore on the adaptations which have occurred to preceding input utterances; and, in turn, those adaptations depend on the recognitions of all the input utterances presented thus far. Thus, the system's response (both of recognition and of adaptation) to any input utterance depends not only on that input utterance itself but on the whole sequence of inputs up to and including it.

An atypical utterance occurring early in the input sequence may cause degradation of a template, so that this template does not closely match subsequent utterances of the word it is meant to represent, and thus fails to be improved by adaptation to them – whereas if the atypical utterance is later in

the input sequence it will have less of a harmful effect on the overall recognition performance, both because of the smaller number of subsequent utterances (on which errors may occur because of the degraded template) and because the template may already have been (beneficially) adapted several times before the atypical utterance is encountered so that it is more robust against unhelpful adaptation. (The latter effect will occur particularly in the case of adaptation with the optimisation system of weighting, where the weight on the input in the averaging is smaller when the template has been previously adapted.) Also, if one word of the vocabulary occurs in the input sequence several times before the first occurrence of another word, the template for the former word is liable to be adapted several times while the latter word's template is still unadapted, and this may affect the overall performance; by contrast, if the input consists of successive repetitions of a fixed sequence containing each word of the vocabulary once, the progress of the adaptation will tend to be nearly uniform across all the words (with the exception of any words whose initial templates correspond too poorly to the input utterances for adaptation to occur, or words which are confusable with other words in the vocabulary and therefore are often not recognised correctly). These effects of input order will cause some variation in the results with supervised adaptation; but in the case with unsupervised adaptation the effects will be magnified because of the possibility of adaptation to wrongly recognised inputs.

The design of experiments to evaluate a recognition system with template adaptation must take this sensitivity to input ordering into account. The variability of the results according to particular input utterances and their positions in the input sequence introduces a high level of statistical "noise" into the evaluation. To obtain statistically significant results, therefore, it is necessary either to compensate for the fluctuations (for instance by use of the same data in

the same order for the testing of different sets of system parameters, or of differently adapted templates, so that in the comparative results the effects of particular input utterances or sequences are cancelled out) or to overcome them by averaging over a large number of trials with different input sequences. In practice, to obtain reliable results without requiring an excessive amount of data and computation, both of these techniques may have to be combined.

Another source of variability in the results, which is common to all experimental evaluations of template-based word recognisers, is the choice of the template set. To overcome this, it may be necessary to perform experiments using several template sets and then average the results obtained (as was done in the experiments already described in chapter 4).

The need for appropriate design of experiments to overcome these forms of variability was demonstrated empirically in the course of the work done on template adaptation. The experimental design was therefore developed, as the research progressed, until a satisfactory level of reliability in the results could be attained.

The two main series of experiments (with the four-speaker digits data base) were designed to evaluate the degrees of improvement in the templates attained after specified amounts of adaptation. In the first series of experiments, each trial consisted of recognition of a specified sequence of utterances, using a specified set of (initial) templates, without adaptation and with each of a number of sets of adaptation parameters. The differences in recognition accuracy between the cases with adaptation and the case without adaptation were found, for each successive short section of the input sequence. Each such difference was a measure of the improvement in the templates' recognition performance attained through adaptation up to the section of the input sequence on which it was computed. (This improvement could alternatively have been measured by

comparing the results on different (earlier and later) subsequences of the input during the adaptive recognition. However, the use of differences between results (with and without adaptation) on the same input subsequence has the advantage that it eliminates much of the variability in recognition accuracy due to characteristics of the particular utterances in the different input subsequences.) These differences were computed for a number of trials, for the different speakers and (for each speaker) for different template sets and input sequences. Statistics (means and the corresponding standard error estimates) of these differences were used to evaluate the effects of the adaptation (according to the procedure set out in the appendix).

In the second series of experiments, a slightly more complicated procedure was adopted. This had two phases – an adaptation phase, in which a sequence of words was recognised with template adaptation, and at certain points in the sequence of input (after specified numbers of recognitions) the adapted templates were stored; and an evaluation phase, in which a standard set of input data (different from the adaptation input) was recognised using each of the stored template sets (including the original set of unadapted templates) in turn, without any further adaptation. In this case, the evaluation of the adaptation was based on the results obtained in the second phase using the adapted templates on the standard set of evaluation data. This two-phase procedure has the advantage that the results after different amounts of adaptation are more strictly comparable, being based on recognition of the same set of evaluation data, rather than on the improvements over non-adaptive recognition on different input subsequences. Because the recognition in the evaluation phase is without continuing adaptation, the evaluation data sequence can contain any number of utterances, without loss of resolution on the amount-of-adaptation scale (such as would be occasioned by the use of results on longer subsequences

in the one-phase procedure used in the first series of experiments). Having a large number of utterances in the evaluation data set reduces the level of random variability in the evaluation phase. As in the first series, results were obtained for a number of template sets and input sequences for each speaker, and means and standard error estimates (of the improvements due to adaptation, and of the actual recognition accuracies) were computed over all the trials. The number of trials required was smaller with the two-phase procedure, however, because of the reduced variability of the results from individual trials.

Because of the significance of the input ordering in adaptive recognition (discussed above), and the limited amount and fixed word order of the data collected in the interactive sessions, in most of the experiments random permutations were applied to the original chronologically-ordered data to obtain the input sequences. This allowed different input sequences to be constructed from the same set of utterances. A different random reordering was used for each trial, but within each trial the same randomly ordered sequence was preserved for recognition using all the different sets of adaptation parameters.

With this random reordering of the input data, detailed chronological information is lost. If the randomised input sequence consists of utterances from a single interactive session, then any characteristics of the session as a whole are retained – and thus the difference between intra-session recognition (with templates formed from utterances collected on the same occasion) and inter-session recognition (with templates from a different session) can be evaluated – but any systematic variations with time during the session will be dispersed by the reordering and will be indistinguishable from random utterance-to-utterance variations. And if the randomised sequence includes utterances from two or more sessions on different occasions, both within-session changes and session-to-session differences will be dispersed. Thus procedures with randomised input

ordering do not allow realistic evaluation of the tracking form of adaptation, which is designed to track those gradual changes and session-to-session differences which are blurred completely by the randomisation. Accordingly, in most of the experiments, which used random ordering of the input, only the optimisation form of adaptation was evaluated.

(Some temporal information could be retained in a randomised reordering of input by restricting the randomisation to groups of utterances close together in time – for instance, by randomising each short subsequence of the utterances within a session, and then concatenating the subsequences in their chronological order; or, in the multiple-session case, by randomising the order of the utterances within each session and then concatenating the randomised single-session sequences. This limited randomisation would entail loss only of local (short-term) temporal information: any longer-term effects would be preserved. However, the effectiveness of the randomisation in overcoming the effects of particular utterances and of a fixed word order would be reduced by these limitations. In particular, after randomisation each subsequence would still contain the same number of occurrences of each word of the vocabulary as before (unless the random rearrangement was accompanied by the omission of some utterances randomly selected from the sequence).)

In addition to the main series of experiments, some more limited experiments were conducted to explore particular aspects of the adaptation parameters. These included some experiments with data collected by input of randomly generated word sequences during the interactive sessions; in these cases, unlike those with repetitions of a standard sequence of words as input, more realistic evaluation of the tracking form of adaptation was possible, since the distribution of words of the vocabulary in the chronologically ordered input sequence had a more natural degree of variability. (The results were less general than those

from the main series of experiments, however, in that the input was from only one speaker). In these experiments, the one-phase procedure, with comparison of subsequence results with and without adaptation, was employed, as in the first series of experiments with the four-speaker data base. In some cases only the overall recognition accuracies, without adaptation and with the adaptation parameters under consideration, were measured; this is the simplest case of the one-phase procedure, in which there is only one subsequence, consisting of the whole input sequence, in each trial.

6.3.2: Details of experiments and results

The subsections below (sections 6.3.2.1 to 6.3.2.5) contain details of several series of experiments conducted to investigate aspects of template adaptation in speaker-specific isolated word recognition. In each section, the results of the experiments described are tabulated, and features of these results are discussed. The results presented relate mostly to the effects of adaptation on recognition accuracy; but section 6.3.2.5 includes also some results as to its effects on the computation required per recognition in the three-stage comparison procedure. Some results (with adaptation but no compensation) have been published previously, in the third paper [258] attached at the end of this thesis. Further results (as in sections 6.3.2.1 and 6.3.2.5 below) are included in another paper [260] whose publication is anticipated during 1988 or 1989.

The results of some early adaptation experiments (with the tracking form of adaptation weighting, and no compensation) with a different data base are given in the second of the attached papers [255]. The recognition accuracies obtained in these experiments were substantially poorer than those achieved in the later experiments, because of errors in the LPC analysis program used during the col-

lection of the earlier data. All the experiments described below were conducted with data processed using the corrected analysis.

6.3.2.1: Estimation of compensation factors

The initial estimates of the compensation factors to be applied to the distances for adapted templates (as mentioned in section 5.3) were obtained from an examination of the ratios of correct-word distances (on the same input utterances) before and after adaptation, for several sets of digit templates. (The adaptation used incorporated the tracking system of weighting, with input weight w_0 equal to 0.2.) The compensation factor for a template adapted n times, for each value of n from 1 to 4, was made approximately equal to the average ratio of the distance before adaptation to the distance after n adaptations of the template. The distances and ratios were found for all three comparison stages used in the decision procedure. The ratios tended to be slightly smaller (closer to 1.0) for the second and third stages than for the simple first-stage comparison, implying that the distances obtained at these later stages were altered less by adaptation of the templates. However, it was not thought to be worthwhile (in view of the increased complexity of the system parameters that would result) to specify different compensation factors for distances at the three stages, and so, for each class of initial templates, only a single factor was determined for each value of n. Three classes of initial template sets were defined: unadapted templates (formed from utterances on a separate occasion), adapted templates (stored after a previous recognition session) and new templates (formed from training utterances provided immediately before the input to be recognised). The compensation factors used for these three classes of initial templates were designated "u", "a" and "n" respectively. The factors for values of n exceeding 4

were extrapolated to continue the trends observed in the factors for the values of n up to 4. These compensation factors, and others which were defined later, are listed in table 6.1. Compensation "a" was adopted for the interactive recognition sessions, using adapted templates, in which the digits data base was collected; for some of the later interactive sessions this was replaced by compensation "c".

Recognition tests with various compensation factors were conducted on input sequences collected from speaker 1, using several initial template sets in each case. The sequences of words collected for these experiments were constructed so as to include instances where one word occurred several times before the first occurrence of another word, as might happen in most practical applications of an isolated word recognition system. The data consisted of two repetitions (used separately in the experiments, and denoted by d1a and d1b) of a 160-digit sequence obtained from a statistical table, containing different numbers of occurrences of the different digits; a randomly ordered 200-digit

Table 6.1: compensation factors for speaker-specific template adaptation

Compensation code	Number of adaptations 1	2	3	4	5	6	7	8
1	1.1	1.15	1.18	1.19	1.2			
a	1.035	1.07	1.09	1.11	1.12			
b	1.03	1.05	1.07	1.08	1.09			
c	1.02	1.035	1.05	1.06	1.07			
d	1.055	1.1	1.14	1.175	1.2	1.22		
e	1.08	1.15	1.21	1.26	1.29	1.31	1.33	
f	1.11	1.2	1.265	1.365	1.395	1.415	1.43	
g	1.12	1.22	1.29	1.35	1.39	1.43	1.45	1.46
h	1.14	1.25	1.32	1.385	1.43	1.475	1.5	1.51
i	1.2	1.3	1.36	1.4	1.43	1.475	1.5	1.51
j	1.16	1.26	1.33	1.39	1.44	1.47	1.5	1.51
k	1.18	1.26	1.33	1.39	1.44	1.475	1.5	1.51
n	1.07	1.13	1.19	1.25	1.28	1.3		
u	1.1	1.185	1.25	1.31	1.35	1.38	1.4	

sequence (d2) containing each digit 20 times; a sequence of 300 words from the 50-word "t" vocabulary (t1), based on a list of sunrise and sunset times, and containing different numbers of occurrences of different words (in particular, each month name occurred only once); another 300-word sequence from the "t" vocabulary (t2), devised so as to include each word six times; and a randomly generated 1000-word sequence (t3) containing differing numbers of occurrences of the words in the "t" vocabulary. In each case, the utterances for use in the adaptation and compensation experiments were selected from the input from the interactive recognition (as listed in section 6.5) by the method described in section 6.2.4, and there was no subsequent reordering of the sequence. (Thus the tracking form of adaptation could be tested, as well as the optimisation form.)

The results of these experiments are given in table 6.2. For each input data sequence, results are shown for different classes of initial template sets. In the "Templates" column, "1" stands for unadapted single-token templates, "2" for unadapted two-token templates and "a" for adapted templates; the number in brackets following this template class indicator is the number of template sets whose results were averaged to obtain the figures tabulated. The results marked with an asterisk (to the left of the adaptation parameters) were obtained using threshold settings $t_1 = 1.4$ and $t_2 = 1.15$ in the three-stage comparison; those without an asterisk are with $t_1 = 1.6$ and $t_2 = 1.2$. In the "Adaptation" column, "trk" stands for the tracking formulation, and "opt" for the optimisation formulation; the number following this is the weight on the input in the first positive adaptation of each template; and the second number is the weight on the input in negative adaptation (if any). The adaptation was supervised in each case. The sets of compensation factors are listed in increasing order across the table.

Table 6.2: results of adaptation compensation experiments

Input (words)	Templates	Adaptation	Compensation factors none	c	b	a	d	n	e	u	f
d1a	1 (5)	none	97.6								
(160)		trk .2 -.05	99.6			99.6		99.6		99.5	
	2 (1)	none	100.0								
		trk .2 -.05	99.4			99.4		99.4		99.4	
d1b	1 (5)	none	95.25								
(160)		trk .2 -.05	99.1			99.0		98.75		98.6	
	2 (1)	none	98.1								
		trk .2 -.05	99.4			99.4		99.4		99.4	
t1	1 (3)	none	85.8								
(300)		trk .2 -.05	90.3	91.0	90.8	90.9		89.9		88.9	
	a (2)	none	95.0								
		trk .2 -.05	96.5		96.3	95.8		92.5		90.0	
d2	1 (5)	* none	97.0								
(200)		* trk .15		99.2	99.0	99.0	98.9	98.9			
		* trk .2	99.7	99.7	99.6	99.6	99.5	99.5	99.5	99.4	99.4
		* opt .2						98.9			
	2 (1)	* none	97.5								
		* trk .15		99.0	99.0	99.0	99.0	99.0			
		* trk .2	99.5	99.5	99.5	99.5	99.5	99.5	99.5	99.5	99.5
		* opt .2						98.5			
t2	1 (3)	* none	88.8								
(300)		* trk .15	89.6	90.6	90.4	90.6	91.0	91.4			
		* trk .2	89.7	90.4	91.3	91.4	91.7	92.0	92.2	92.7	92.8
		* trk .25						92.1	92.2	92.6	92.7
		* opt .2	89.8	90.7	91.2	91.7	91.9	92.1	92.6	92.2	92.6
	2 (1)	* none	96.7								
		* trk .15	97.3	97.0	97.3	97.3	98.0	97.7			
		* trk .2	96.7	97.0	97.0	98.0	98.7	98.3	98.0	97.3	97.3
		* trk .25						98.0	97.7	97.7	96.7
		* opt .2	96.7	96.7	96.7	97.0	97.7	97.3	97.0	96.7	96.3
t3	1 (3)	* none	85.3								
(1000)		* trk .2	89.0			89.9		91.2		91.5	91.4
		* opt .2	88.9			90.0		89.9		90.7	91.2
	2 (1)	* none	93.1								
		* trk .2	93.6			94.1		94.4		94.2	94.0
		* opt .2	94.1			94.2		94.1		94.4	94.5

The results in table 6.2 for the "t" vocabulary show improvements in recognition accuracy (over the case with adaptation and no compensation) when appropriate compensation factors are applied. In the results on t2 and t3, with single-token initial templates, the increase in accuracy due to the compensation technique is typically greater than 2%. Smaller and less consistent improvements can be seen in the recognition of t1. The confidence – as estimated from the differences in the individual template sets' overall accuracies (computed over all three "t" input sequences) – that compensation "a" is better than no compensation for the "t" vocabulary when single-token initial templates are used, with the tracking form of adaptation and input weight 0.2, is 0.993; the confidence that compensation "n" is better than no compensation is 0.97. The results for the digits show no such improvements: with small compensation factors, the results are the same as with none, and with larger compensation factors the accuracies tend to be reduced slightly. This inconsistency between the two vocabularies could be associated with characteristics of the words they contain, whereby unadapted templates for the "t" vocabulary may be more suboptimal for matching new utterances of the words they represent than unadapted digit templates, so that adaptation can make greater improvements in the distances obtained, and hence a greater degree of compensation is appropriate. However, it should be noted that the numbers of errors on the digits in all cases of adaptive recognition (with and without compensation) are small, and the differences among these cases involve different recognitions of only a very few utterances, so that the comparative results on this vocabulary are not highly significant. Indeed, because of the small numbers of recognition trials conducted (with only a single speaker), and the high degree of variability inherent in adaptive recognition results, all the results in table 6.2 should be treated with caution.

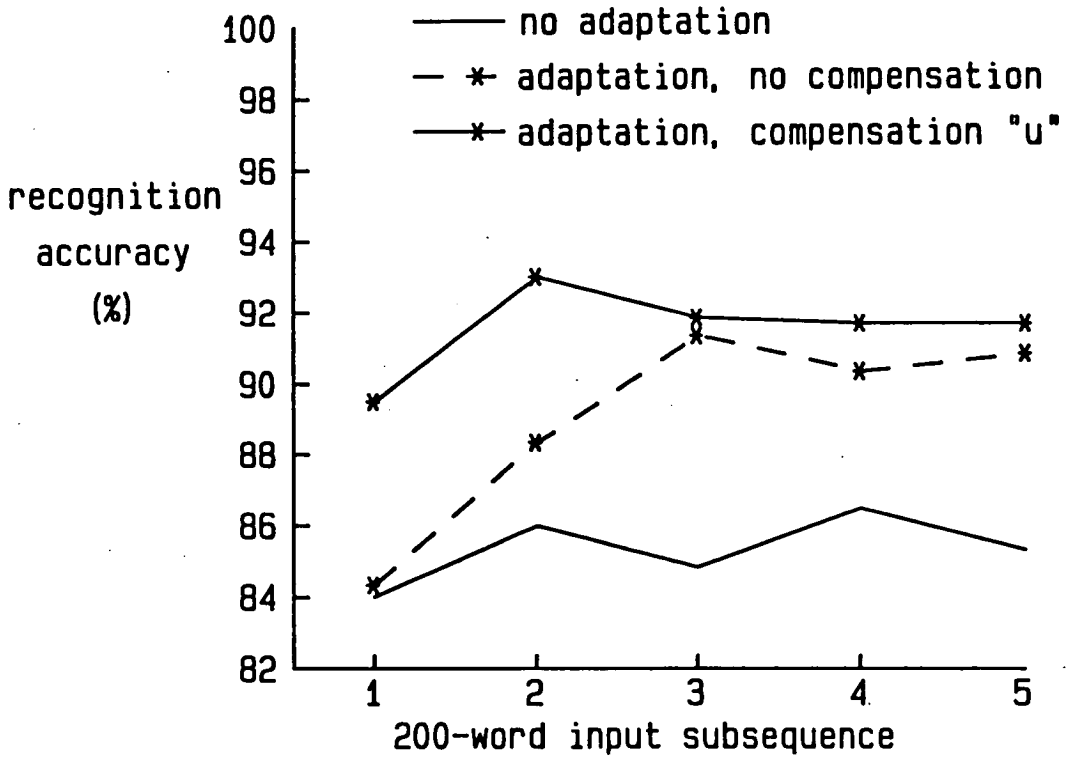
A general tendency apparent in the results is that the optimal compensation factors for two-token or previously-adapted templates are smaller than those for (unadapted) single-token templates. This is as might be expected, since single-token templates will have greater random deviations from the typical forms of the words, and will thus tend to yield larger distances, which can be reduced considerably by adaptation of the templates. It is also evident that the difference between the accuracies with optimal compensation and with none is generally smaller for two-token or previously-adapted templates.

The results with input weight 0.15 in the adaptation are generally poorer than those with input weight 0.2, while those with weight 0.25 are similar to those with weight 0.2. The best results with optimisation weighting are in most cases slightly poorer than the best results obtained with the tracking form. No clear difference in optimal compensation can be seen among the different adaptation weightings, though it might have been expected that the optimal compensation factors (for small numbers of adaptations) would increase as the input weight increased, since each adaptation would make a greater difference to the template. In general, no reliable conclusions about adaptation weighting can be drawn from this limited set of experiments.

Some further comparison of compensation factors was included in the two main series of experiments, in which the larger numbers of trials yielded a higher level of statistical significance in the results. The results obtained are stated in sections 6.3.2.4 and 6.3.2.5.

The results for successive 200-word subsequences of the 1000-word sequence t3, without adaptation, with adaptation (tracking, input weight 0.2) but no compensation, and with adaptation and compensation "u", averaged over the three single-token template sets, are plotted in figure 6.1. It can be seen that almost the same level of accuracy is attained on the last 600 words with adaptation and

Figure 6.1: results for 1000-word sequence from "t" vocabulary with and without adaptation and compensation



no compensation as with adaptation and optimal compensation, but this level is approached more slowly (over the first 400 words) in the case without compensation.

6.3.2.2: Experiments with negative adaptation

Some experiments were conducted with input data sets d2 and t2 (as used for the experiments with compensation factors in section 6.3.2.1) to determine the effects of adjusting the negative weight on misrecognised input for negative adaptation. The results are summarised in table 6.3 (in which the notation

Table 6.3: results of negative adaptation experiments

Input (words)	Templates	Positive adaptation	Compens-ation	Input weight for negative adaptation 0	-.025	-.03	-.04	-.05	-.06	-.07
d2	1 (5)	* trk .2	n	99.5	99.4	99.5	99.5	99.5		
(200)		* opt .2	n	98.9				98.9		
	2 (1)	* trk .2	n	99.5	99.5	99.5	99.5	99.5		
		* opt .2	n	98.5				98.5		
t2	1 (3)	* trk .2	none	89.7				89.9		
(300)		* trk .2	n	92.0	92.2	92.2	92.4	92.2	92.3	92.3
		* opt .2	n	92.1				92.1		
	2 (1)	* trk .2	none	96.7				96.7		
		* trk .2	n	98.3	98.3	98.3	98.3	98.3	98.3	98.3
		* opt .2	n	97.3				97.3		

adopted is similar to that in table 6.2). Negative adaptation was found to make almost no difference to the accuracy on the digit sequence d2 (not surprisingly since there were very few misrecognised utterances to adapt away from), and to yield a small improvement overall on t2, which was essentially due to the improvements attained with one of the three sets of single-token templates. (On other sets of templates, the accuracy was in some cases reduced slightly by the negative adaptation.)

Some further comparison of results with and without negative adaptation is included in section 6.3.2.4.

6.3.2.3: Comparison of alignment options in adaptation

Experiments were conducted with data sets d1a, d1b and t1 (as in section 6.3.2.1) and with another set of "t" data (t4) (as used in the multiple-stage recognition experiments in section 4.4, and described in section 6.2.4), to compare the effects of adaptation with linear alignment and with DTW alignment. (All the experiments with adaptation reported in the preceding sections used DTW

alignment.) The tracking form of adaptation was adopted, with input weight 0.2 in positive adaptation and -0.05 in negative adaptation. No compensation was applied. The elimination thresholds t_1 and t_2 in the comparison procedure were set to 1.6 and 1.2 respectively. The results are given in table 6.4. The same notation for template sets is adopted as in tables 6.2 and 6.3.

The results for the "t" words show a fairly consistent superiority of DTW adaptation to linear adaptation. (For six of the seven "t" template sets used in these experiments, the overall average accuracies were better with DTW adaptation than with linear adaptation; for the seventh, the accuracies with the two alignment options were identical.) The results on the digits show little difference overall between the two forms of alignment, and are less consistent across template sets; the significance of the difference due to alignment in these digit recognition results is very low. It seems likely that the superiority of DTW over linear alignment will in general tend to be greater for vocabularies contain-

Table 6.4: results of comparison of alignment options in adaptation

Input (words)	Templates	Adaptation none	linear	DTW
d1a	1 (5)	97.6	99.25	99.6
(160)	2 (1)	100.0	100.0	99.4
d1b	1 (5)	95.25	99.0	99.1
(160)	2 (1)	98.1	100.0	99.4
t1	1 (3)	85.8	89.7	90.3
(300)	a (2)	95.0	95.3	96.5
t4	1 (3)	89.0	90.3	90.7
(300)	a (1)	90.0	90.7	92.3
	2 (1)	95.3	96.3	96.7

ing long words, since these give more scope for non-linear variations of timescale which may result, with linear alignment, in the averaging together of parts of different repetitions which are phonetically distinct and acoustically very different.

For the main series of experiments, described in the next two sections, and for the experiments with speaker-independent initial templates described in chapter 7, the DTW alignment option was adopted. However, it is worth noting that the linear alignment option requires an order of magnitude less computation than the DTW option, and therefore it might in some circumstances be worthwhile using linear alignment to gain an improvement in the speed of adaptive recognition (at the cost of some loss of accuracy).

6.3.2.4: First main series of adaptation experiments

Experiments were conducted, by the one-phase procedure described in section 6.3.1, with the four-speaker digits database, to measure the improvements in recognition performance (over the case without adaptation) attained during adaptive recognition of sequences of 50 digits.

The first experiments were conducted using the two template sets (of single-token templates and of two-token averaged templates) which had been formed in interactive training sessions for each speaker. For each of the first three speakers, for each template set, each of the 10 50-utterance sequences in turn was used as input for recognition, without adaptation and with each of a number of adaptation parameter settings. The recognition accuracies on the five 10-word subsequences of each input set were found, and the differences between corresponding subsequence accuracies with adaptive recognition and with non-adaptive recognition were derived. The overall accuracies for the 50-word

sequences, and their differences, were also found. For each speaker and template set, the results were averaged over the 10 50-utterance input sequences. These results are summarised in table 6.5.

In the "adaptation" column of the table, "U" stands for unsupervised adaptation, and "S" for supervised; "t" for tracking, and "o" for optimisation; the number following "t" or "o" is the weight on the input utterance in the first adaptation of each template; and the second number is, in the case of unsupervised adaptation, the threshold imposed on the ratio of the best two word distances as a condition for adaptation, or, in the case of supervised adaptation, the weight on the input in negative adaptation. In the cases marked with an asterisk, the endpoint adjustment technique was incorporated in the third stage of the recognition procedure (and in the adaptation). The template elimination thresholds t_1 and t_2 were set to 1.6 and 1.1 respectively. (These threshold values were retained throughout the subsequent experiments described in this section and in section 6.3.2.5.) Compensation "a" (as in table 6.1) was applied in each case. For each input sequence or 10-word subsequence, the results given are the mean increase in recognition accuracy over the case with no adaptation, over the three speakers, and an estimate of the standard error of this mean (considered as an estimate of the mean that would be obtained for a population of many speakers) computed from the variability of the recognition improvements across the speakers. (These experiments were not extended to speaker 4 because it had become evident that more template sets per speaker would be required to improve the reliability of the results.)

One feature of these results is that larger and more consistent improvements in accuracy were attained by adaptation in the case of single-token initial templates than in that of two-token templates. This was to be expected, since

Table 6.5: adaptive recognition results on 50-digit sequences using interactively formed template sets

Template sets	Adaptation	Mean (standard error) of improvement in percentage accuracy over non-adaptive recognition					
		input subsequence					overall
		1	2	3	4	5	
1-token	U t .2 (1.15)	0.00	0.33	1.33	1.00	0.00	0.53
		(0.00)	(0.29)	(0.76)	(0.86)	(1.32)	(0.48)
	U o .2 (1.15) *	-0.33	0.00	0.67	1.33	0.33	0.40
		(0.33)	(0.00)	(0.66)	(0.66)	(1.20)	(0.50)
	S t .2 0	0.00	1.00	1.67	2.67	1.67	1.40
		(0.00)	(0.86)	(0.58)	(0.58)	(0.29)	(0.23)
	S t .2 -.05	0.00	1.00	2.33	3.00	2.33	1.73
		(0.00)	(0.86)	(0.58)	(0.86)	(0.29)	(0.55)
	S o .2 0	0.00	1.00	1.67	2.67	1.67	1.40
		(0.00)	(0.86)	(0.58)	(0.58)	(0.29)	(0.31)
	S o .25 0	0.00	1.33	1.33	3.00	1.33	1.40
		(0.00)	(1.16)	(0.29)	(0.86)	(0.29)	(0.40)
2-token	U t .2 (1.15)	0.00	0.00	1.00	0.00	0.67	0.33
		(0.00)	(0.00)	(0.58)	(0.58)	(0.66)	(0.18)
	U o .2 (1.15)	0.00	0.00	1.00	-0.67	0.67	0.20
		(0.00)	(0.00)	(0.58)	(0.88)	(0.66)	(0.23)
	U o .2 (1.15) *	0.33	0.00	1.33	0.33	1.00	0.60
		(0.33)	(0.00)	(0.33)	(0.88)	(1.00)	(0.42)
	S t .2 0	0.00	0.00	1.00	0.00	0.33	0.27
		(0.00)	(0.00)	(0.58)	(0.58)	(0.88)	(0.24)
	S t .2 -.05	0.00	0.00	1.00	0.00	0.67	0.33
		(0.00)	(0.00)	(0.58)	(0.58)	(1.20)	(0.29)
	S o .2 0	0.00	0.00	1.00	-0.33	0.33	0.20
		(0.00)	(0.00)	(0.58)	(0.66)	(0.88)	(0.23)
	S o .25 0	0.00	0.00	1.33	-0.33	0.33	0.27
		(0.00)	(0.00)	(0.33)	(0.66)	(0.88)	(0.29)

the accuracies without adaptation were poorer for the single-token templates (averaging 94.2% over the three speakers) than for the two-token templates (which had an average accuracy of 97.8%), thus leaving more room for improvement.

To improve the reliability of the results, a revised experimental design was adopted. For each of the four speakers, 10 template sets were defined, each

consisting of the first occurrences of all the digits in one of the 10 50-digit sequences. For each of these template sets, each of the remaining nine 50-digit sequences was used as an input sequence. Thus, for each speaker, the results could be averaged over 90 trials (instead of only 10 as in the experiments without multiple template sets). The results are given in table 6.6. The notation for adaptation parameters is the same as in table 6.5; again the compensation factors in each case were those denoted by "a" in table 6.1. The improvements over non-adaptive recognition, for 10-word subsequences and overall, are represented by their means and standard error estimates, computed over the four speakers. (These results are not directly comparable with those in table 6.5, because of the inclusion of speaker 4.) The mean and standard error for the overall recognition accuracy are also given, for non-adaptive recognition and for each case of adaptive recognition.

The standard error figures in table 6.6, especially those for the overall improvements on the 50-digit sequences, are mostly smaller than the corresponding standard errors (for single-token templates) in table 6.5. This is partly because of the larger number of speakers (since the expected standard error of a mean of n samples from a given population is inversely proportional to $\sqrt{n}$: here n is 3 for the results in table 6.5, and 4 for those in table 6.6); but also because the variability is reduced by using a larger number of trials (with different template sets) for each speaker. The overall trends in the results are similar to those in table 6.5: the improvement over the non-adaptive case tends to increase with successive 10-word input sequences, and the improvements are generally greater for supervised adaptation than for unsupervised. There is also a slight difference between the results (with supervised tracking adaptation) with and without negative adaptation, which is consistent across the four speak-

Table 6.6: adaptive recognition results on 50-digit sequences averaged over 10 template sets per speaker

Adaptation	Mean (standard error) of improvement in percentage accuracy over non-adaptive recognition						Mean (s.e.) overall recognition accuracy
	input subsequence					overall	
	1	2	3	4	5		
none							95.633
							(0.69)
U t .2 (1.15)	-0.20	0.56	1.28	1.17	1.03	0.766	96.400
	(0.15)	(0.15)	(0.17)	(0.34)	(0.40)	(0.13)	(0.56)
U o .2 (1.15)	0.00	0.36	1.00	0.72	0.78	0.572	96.205
	(0.08)	(0.07)	(0.32)	(0.25)	(0.25)	(0.07)	(0.67)
S t .2 0	-0.20	0.86	1.89	1.75	2.00	1.261	96.894
	(0.14)	(0.14)	(0.24)	(0.46)	(0.52)	(0.19)	(0.51)
S t .2 -.05	-0.11	1.02	2.03	1.92	2.19	1.411	97.044
	(0.14)	(0.21)	(0.29)	(0.54)	(0.62)	(0.24)	(0.45)
S o .2 0	-0.20	0.92	1.83	1.67	1.86	1.217	96.850
	(0.14)	(0.12)	(0.23)	(0.38)	(0.44)	(0.16)	(0.55)
S o .25 0	-0.20	1.00	2.11	1.81	1.92	1.328	96.961
	(0.15)	(0.20)	(0.16)	(0.41)	(0.63)	(0.19)	(0.52)

ers; the average difference in the overall results is 0.15%, and the estimated standard error of this difference is 0.054, so that the confidence that negative adaptation improves the recognition is 0.97. The other differences, between the results with tracking and with optimisation, and between those with different initial input weights (0.2 and 0.25) in the optimisation case, are likewise of low significance. The results with adaptation on the first few words of each sequence (input subsequence 1) tend to be poorer than without adaptation, as indicated by the negative entries in the first column of results; this suggests that the compensation factors used were not optimal.

Some additional non-adaptive recognition tests were conducted, using the 10 templates from each session to recognise the remaining 40 words from the same session, so as to obtain a comparison of within-session recognition (templates and input from the same session) and cross-session recognition (templates

from one session and input from another – as in the preceding experiments). The average within-session recognition accuracy was 96.25%; this was 0.55% higher than the average cross-session accuracy on the same input data, but this difference is not very significant (its standard error, assessed from the distribution of the single-speaker average differences, being 0.49, which yields confidence 0.83).

Although the variability of the results is reduced, and thus the reliability of the comparative results is improved, by the use of multiple template sets, there remain some fluctuations from one subsequence number to another. For instance, for each set of adaptation parameter values, the average improvement on the fourth 10-word subsequence in table 6.6 is smaller than that on the third subsequence. This could be an effect of the fixed order of the input utterances (in each set of 50) used for all the 10 template sets. To eliminate this effect, a further refinement of the experimental procedure was adopted. For each 50-utterance input set, nine random permutations of the 50 digits were defined. (These were different for each input set.) Before recognition of the input utterances using the nth of the nine template sets derived from the other data sets, the nth of the nine permutations was applied to rearrange the input sequence.

The results with the randomly ordered input sequences are presented in table 6.7. The notation for adaptation parameters is the same as in tables 6.5 and 6.6. The results in the first few lines of table 6.7 are those obtained with the same combinations of adaptation and compensation parameters as in the preceding experiments without random ordering. The remaining results are those with fixed sets of adaptation parameters (unsupervised and supervised adaptation, incorporating the optimisation weighting in each case) and various sets of compensation factors. (As was noted previously, randomly reordered input sequences do not permit realistic evaluation of the tracking form of

Table 6.7: adaptive recognition results on randomly ordered 50-digit sequences (averaged over 10 template sets per speaker)

Adaptation and compensation		Mean (standard error) of improvement in percentage accuracy over non-adaptive recognition						Mean (s.e.) overall recognition accuracy
		input subsequence					overall	
		1	2	3	4	5		
S t .2 0	a	0.16	0.86	1.17	1.48	2.22	1.178	96.811
		(0.10)	(0.18)	(0.23)	(0.31)	(0.16)	(0.10)	(0.59)
S t .2 -.05	a	0.16	0.89	1.20	1.70	2.33	1.256	96.889
		(0.10)	(0.16)	(0.19)	(0.31)	(0.27)	(0.12)	(0.57)
S o .2 0	a	0.22	0.95	1.14	1.34	2.14	1.155	96.789
		(0.16)	(0.13)	(0.21)	(0.27)	(0.19)	(0.07)	(0.62)
S o .25 0	a	0.19	0.89	1.17	1.59	2.22	1.211	96.844
		(0.09)	(0.14)	(0.22)	(0.31)	(0.14)	(0.11)	(0.59)
U t .2 (1.15)	a	0.03	0.33	0.42	0.97	1.58	0.667	96.300
		(0.16)	(0.16)	(0.19)	(0.22)	(0.14)	(0.04)	(0.71)
U o .2 (1.15)	a	0.08	0.36	0.34	0.56	1.00	0.467	96.100
		(0.13)	(0.19)	(0.16)	(0.39)	(0.31)	(0.15)	(0.70)
S o .25 -.05	none	-0.14	0.22	0.44	0.67	1.47	0.533	96.167
		(0.09)	(0.29)	(0.45)	(0.20)	(0.19)	(0.17)	(0.63)
S o .25 -.05	a	0.22	0.86	1.22	1.59	2.25	1.228	96.861
		(0.07)	(0.09)	(0.23)	(0.31)	(0.16)	(0.12)	(0.58)
S o .25 -.05	d	0.45	1.11	1.41	1.89	2.55	1.483	97.117
		(0.14)	(0.20)	(0.32)	(0.26)	(0.25)	(0.15)	(0.54)
S o .25 -.05	n	0.39	1.28	1.67	2.03	2.72	1.617	97.250
		(0.11)	(0.30)	(0.32)	(0.33)	(0.33)	(0.20)	(0.49)
S o .25 -.05	u	0.47	1.53	1.81	2.08	2.75	1.728	97.361
		(0.16)	(0.36)	(0.41)	(0.32)	(0.27)	(0.22)	(0.48)
S o .25 -.05	f	0.45	1.53	1.86	2.14	2.78	1.750	97.383
		(0.19)	(0.36)	(0.35)	(0.34)	(0.30)	(0.21)	(0.49)
S o .25 -.05	g	0.50	1.45	2.00	2.14	2.81	1.778	97.411
		(0.19)	(0.38)	(0.41)	(0.32)	(0.33)	(0.24)	(0.46)
S o .25 -.05	h	0.42	1.53	2.03	2.20	2.83	1.799	97.433
		(0.09)	(0.38)	(0.49)	(0.30)	(0.32)	(0.26)	(0.46)
U o .2 (1.15)	none	-0.34	-0.64	-0.61	-0.58	-0.56	-0.544	95.089
		(0.14)	(0.20)	(0.16)	(0.26)	(0.14)	(0.11)	(0.72)
U o .2 (1.15)	d	0.05	0.67	0.70	1.02	1.59	0.805	96.439
		(0.09)	(0.08)	(0.14)	(0.32)	(0.34)	(0.13)	(0.65)
U o .2 (1.15)	n	0.08	0.81	0.95	1.28	1.97	1.016	96.650
		(0.09)	(0.16)	(0.17)	(0.30)	(0.19)	(0.10)	(0.63)
U o .2 (1.15)	u	0.11	0.86	1.28	1.56	2.08	1.178	96.811
		(0.13)	(0.20)	(0.32)	(0.20)	(0.14)	(0.11)	(0.59)
U o .2 (1.15)	f	0.16	0.86	1.28	1.64	2.06	1.200	96.834
		(0.09)	(0.20)	(0.34)	(0.20)	(0.17)	(0.12)	(0.58)
U o .2 (1.15)	g	0.16	0.86	1.22	1.64	2.08	1.194	96.827
		(0.13)	(0.20)	(0.34)	(0.16)	(0.25)	(0.12)	(0.57)
U o .2 (1.15)	h	0.05	0.84	1.27	1.61	2.03	1.161	96.794
		(0.14)	(0.26)	(0.39)	(0.15)	(0.16)	(0.18)	(0.52)

adaptation; but the randomised word order does allow a better comparison of different settings of the compensation factors than a fixed word order would.) For each set of adaptation parameters, the results with different compensation factors are listed in increasing order of the amount of compensation.

The results without adaptation on the randomly ordered input are identical to those in table 6.6: in non-adaptive recognition, the order of the input utterances makes no difference to the overall recognition accuracy.

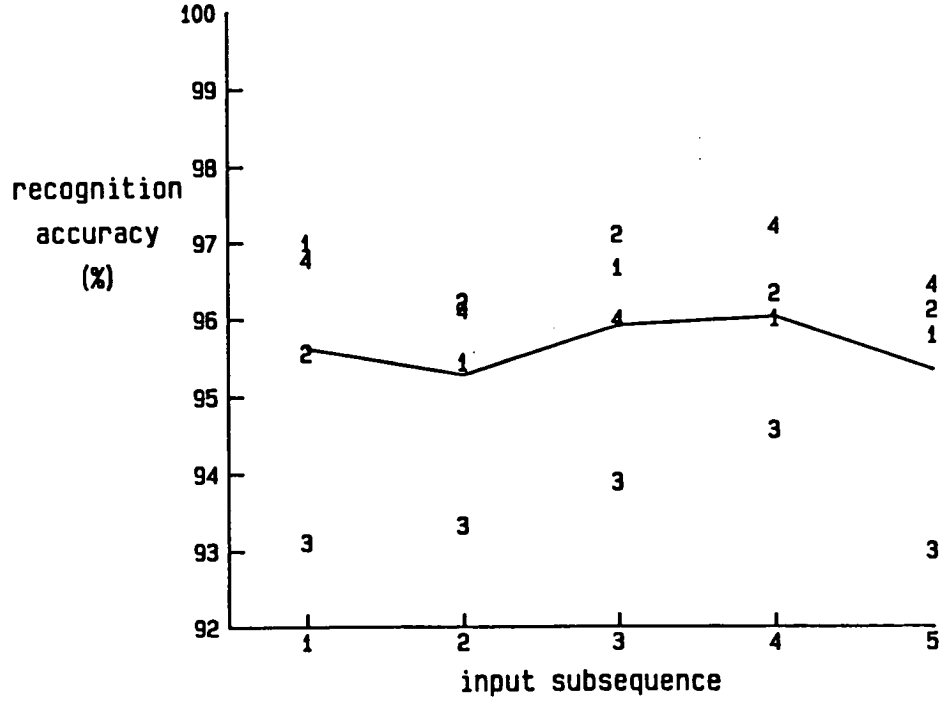
The results with compensation "a" in table 6.7 are broadly similar to the corresponding results in table 6.6. The main differences are, firstly, that the overall improvements in recognition accuracy are smaller for the randomly ordered input, and, secondly, that the irregular variations across input subsequences, and the standard errors (estimated from the variations across speakers) for individual subsequence numbers, are reduced by the random reordering. The first of these differences is accentuated by the fact that the compensation factors have values smaller than the optimal ones: with optimal compensation factors, the advantage of the more regular word order over the random orders should be smaller. The reduction of the level of variability in subsequence results, due to the use of multiple random orderings of the input data, allows the underlying trend of increasing improvement with successive subsequences to be seen more clearly in table 6.7 than in table 6.6: in table 6.7 the improvement due to adaptation nearly always increases from one subsequence number to the next. The difference between corresponding supervised adaptation results with and without negative adaptation is smaller than in table 6.6, but still fairly significant: the mean difference is 0.077%, and the standard error is 0.03, yielding the conclusion that negative adaptation improves recognition with confidence 0.96.

The results with different compensation factors reveal the importance of appropriate compensation for optimal performance. With no compensation, the recognition of randomly ordered input is consistently degraded (relative to the case without adaptation) by unsupervised adaptation, and is improved by only about 0.5% over the 50-word input sequence by supervised adaptation. With optimised compensation, an improvement of 1.2% is attained using unsupervised adaptation, and the improvement with supervised adaptation is increased to 1.8%. (The optimal compensation factors are greater for the supervised adaptation case than for the unsupervised, probably because the weight assigned to the input, relative to the weight on the initial template, in the adaptation is greater; but the differences in performance among compensation settings "u", "f", "g" and "h" are small, and of low significance, in each case.) The average improvement on the fifth 10-word input subsequence (with optimal compensation) is about 2.1% using unsupervised adaptation, and 2.8% using supervised adaptation. This level of improvement is approached more gradually, over the earlier subsequences, with unsupervised adaptation than with supervised adaptation. (This may be partly an effect of the larger weight on the input in the supervised adaptation in these experiments; but the same difference in the rates of progress of unsupervised and supervised adaptation can be seen in the results with compensation "a" where the same weighting was used for both. There is a case for keeping the input weight small in unsupervised adaptation because of the possibility of adaptation to wrongly recognised inputs.) The improvements, on the fifth subsequences and overall, due to adaptation (supervised or unsupervised) with optimal compensation are highly significant: the confidence that the adaptation improves the recognition ranges from 0.997 to 0.9994.

The variation across speakers in the actual percentage recognition accuracy obtained is reduced by adaptation; and more so as the compensation factors are

improved – whereas the variation across speakers in the improvement over the non-adaptive case is not generally reduced. This is particularly evident in the supervised adaptation results. Here, with no compensation, the standard error of the recognition accuracy (as estimated from the variation across speakers) is 0.63, and the standard error of the improvement over non-adaptive recognition is 0.17; with compensation "n", these figures are reduced and increased (respectively) to 0.49 and 0.20; with compensation "h", they become 0.46 and 0.26. This indicates that, when the adaptation and compensation settings are optimised, the improvement in recognition accuracy due to the adaptation tends to be greater for a speaker whose recognition rate without adaptation was poorer – which is a desirable feature of the adaptive system.

Figure 6.2: results for 50-digit sequences without adaptation



The results for the individual speakers, together with the averaged results, are plotted in figures 6.2 to 6.6, for the case without adaptation and for the cases of supervised and unsupervised adaptation, without compensation and with optimised compensation. In each of these figures, the results marked by the numbers from 1 to 4 are those for the respective speakers, and the line represents the results averaged over the speakers. It can be seen from a comparison of these figures that the improvements, with adaptation and optimised compensation factors, over the case without adaptation, and over the case with adaptation but no compensation, are consistent across the four speakers. The plots also confirm that the improvement due to adaptation (with optimal compensation) is greater for a speaker with poor non-adaptive performance: without adaptation, the recognition of speaker 3's utterances was poorer (averaging 93.58%) than the recognition for the other speakers (whose average accuracies ranged from 96.18% to 96.51%), but with adaptation this gap was narrowed, and the respective overall accuracies were 96.13% and 97.60%-98.29% in the case of supervised adaptation, and 95.04% and 97.36%-97.44% with unsupervised adaptation.

6.3.2.5: Second main series of adaptation experiments

Further experiments were conducted, with the same four-speaker digits data base, using the two-phase procedure described in section 6.3.1, to measure the effects of adaptation over longer sequences of input utterances. As in the preceding experiments, 10 template sets were used for each speaker, derived from the first occurrences of the digits in the 10 50-word sequences. For each template set, the input data for adaptive recognition consisted of the 450 utterances (45 of each digit) from the other nine 50-word sequences. In each trial,

Figure 6.3: results for 50-digit sequences with supervised adaptation and no compensation

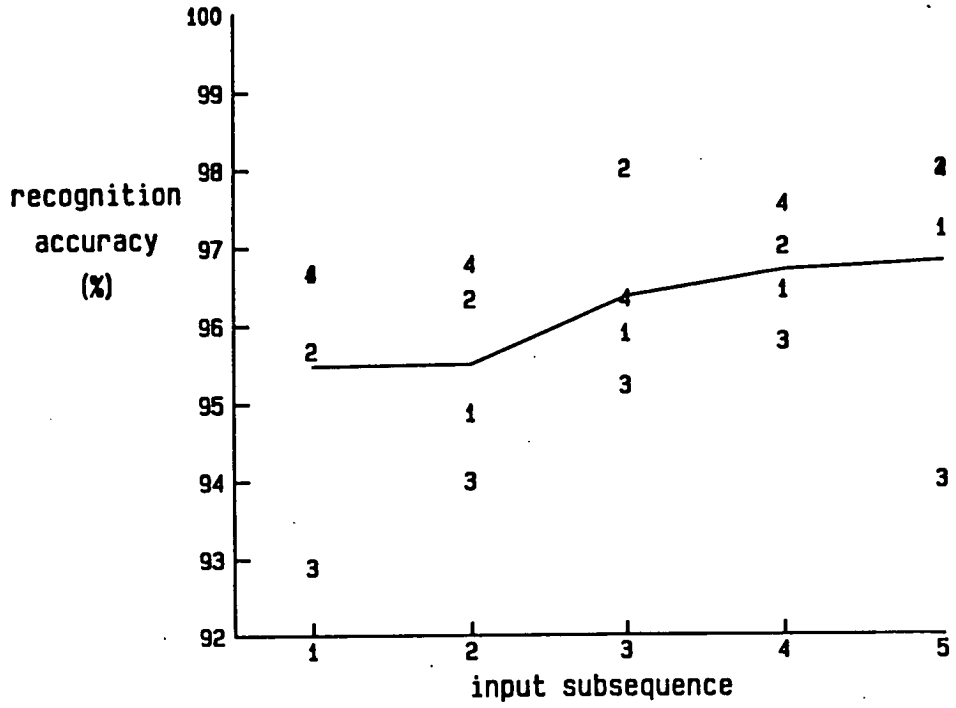


Figure 6.4: results for 50-digit sequences with supervised adaptation and compensation "h"

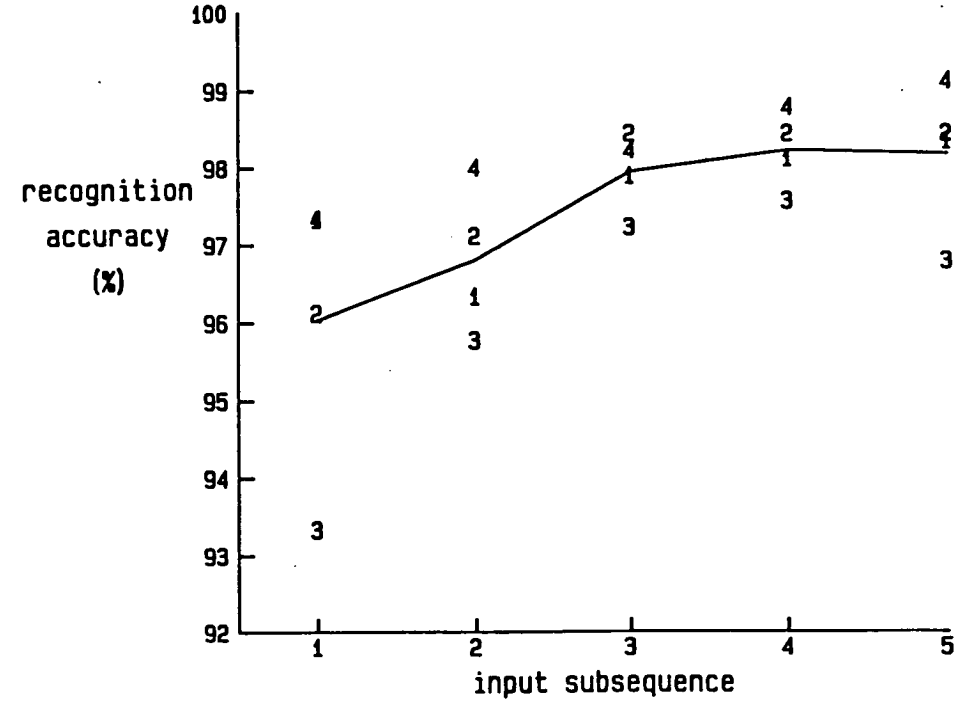


Figure 6.5: results for 50-digit sequences with unsupervised adaptation and no compensation

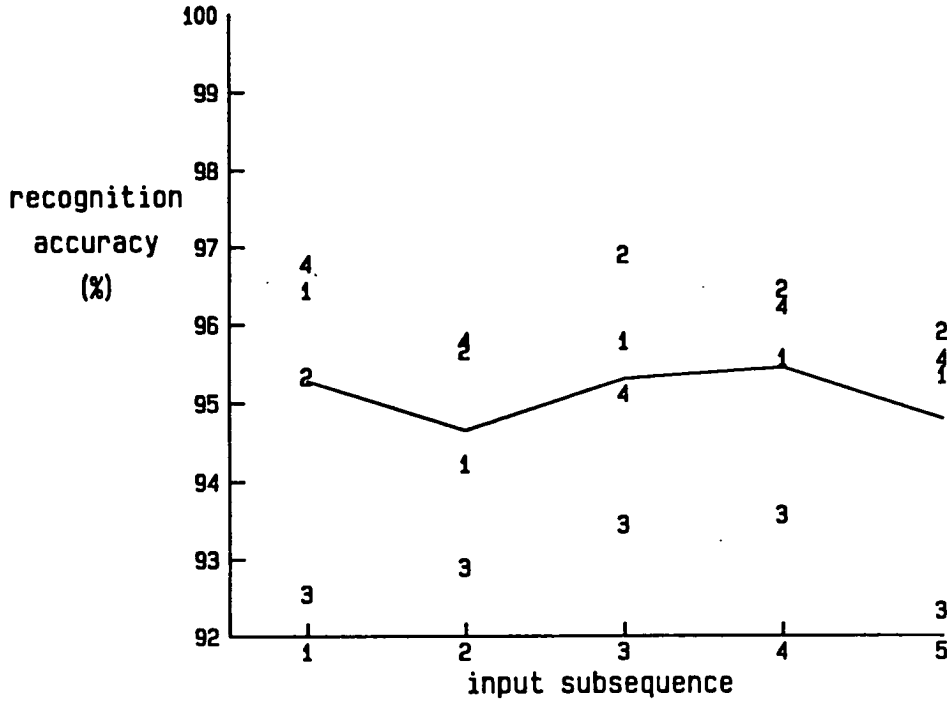
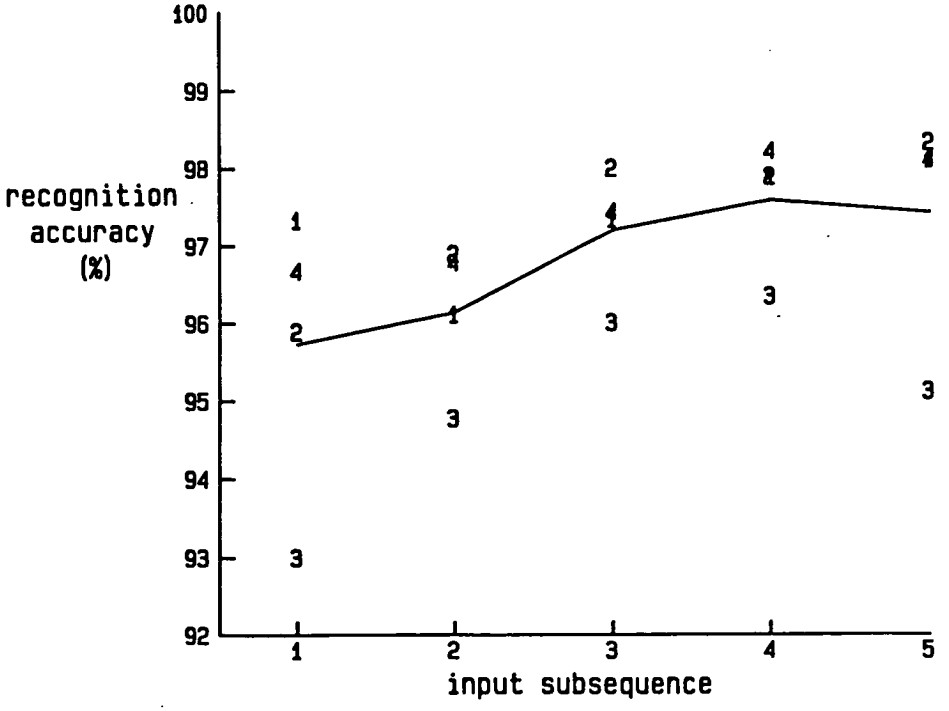


Figure 6.6: results for 50-digit sequences with unsupervised adaptation and compensation "u"



the 450 utterances were randomly reordered and recognised with and without adaptation; during the adaptive recognition, the adapted templates were stored after every multiple of 10 recognitions up to 100, and also after 150, 200 and 250 recognitions; and the improvements in the templates were evaluated by using each adapted template set (and the original unadapted template set) to recognise the last 200 input utterances. This procedure was applied twice for each set of initial templates, with two different random orders of the same 450 input utterances. (Different random orders were used for all the different initial template sets.)

In each trial, several sets of adaptation (and compensation) parameter values were tested, including both supervised and unsupervised adaptation. The optimisation form of weighting was adopted in every case, because the experimental design with random reordering of the input data was unsuitable for the evaluation of the tracking form. The sets of parameter values included two of those used in the previous experiments (those denoted in table 6.7 by "S o .25 -.05 u" and "U o .2 (1.15) u"), and also some others with larger input weights (or equivalently, smaller weights on the initial templates).

During the adaptation phase of each trial, a list of adaptation counts was stored for use with each set of adapted templates: for each template, the number stored was the number of times that template had been (positively) adapted so far. In the evaluation phase, these adaptation counts were read in along with the templates, and compensation factors were assigned to the templates accordingly.

In these experiments, the main evaluation was based on the results of the (non-adaptive) recognition of the last 200 input utterances, following the adaptive recognition phase. However, the recognition results obtained on 10-utterance subsequences during the adaptive recognition were also computed.

These allow comparison of the effects of adaptation in these experiments with those in the first series of experiments (as presented in table 6.7). The results on the first 10 10-word subsequences, and on the full 450-word sequences, are given in table 6.8.

A comparison of the results for the first five 10-digit subsequences in table 6.8 with the corresponding results in table 6.7 reveals that the improvements in table 6.8 are consistently smaller. There are two factors which may contribute to this effect.

Firstly, the results in table 6.8 are for input sequences composed of utterances from nine different interactive sessions, rather than single-session input

Table 6.8: adaptive recognition results on randomly ordered 450-digit sequences (averaged over 10 template sets per speaker)

Adaptation and compensation	Mean (standard error) of improvement in percentage accuracy over non-adaptive recognition (10-digit) input subsequence										overall	Mean (s.e.) overall recognition accuracy
	1	2	3	4	5	6	7	8	9	10		
none												95.633 (0.688)
S o .25 -.05 u	0.38	0.75	1.25	1.25	1.63	2.13	2.00	3.25	2.63	2.88	2.684	98.317
	(0.43)	(0.60)	(0.66)	(0.66)	(0.38)	(0.32)	(0.94)	(0.48)	(0.69)	(0.13)	(0.352)	(0.340)
S o .5 -.05 h	0.00	0.13	1.50	1.62	1.75	1.38	2.13	3.25	2.62	2.50	2.691	98.325
	(0.36)	(0.65)	(0.36)	(0.87)	(0.43)	(0.52)	(0.63)	(0.48)	(0.78)	(0.29)	(0.285)	(0.404)
S o .5 -.05 j	-0.13	-0.38	1.50	1.50	1.50	1.25	1.88	3.25	2.88	2.37	2.636	98.270
	(0.13)	(0.43)	(0.21)	(0.82)	(0.54)	(0.43)	(0.85)	(0.48)	(0.72)	(0.24)	(0.305)	(0.386)
S o .5 -.05 i	-0.38	0.00	1.50	2.00	1.75	1.50	2.00	3.25	2.75	2.50	2.633	98.267
	(0.38)	(0.89)	(0.21)	(0.64)	(0.43)	(0.54)	(0.64)	(0.48)	(0.83)	(0.29)	(0.315)	(0.382)
U o .2 (1.15) u	-0.13	0.38	0.88	1.00	1.13	1.50	0.87	2.87	2.25	2.00	2.236	97.869
	(0.24)	(0.38)	(0.38)	(0.73)	(0.24)	(0.21)	(0.52)	(0.56)	(0.78)	(0.45)	(0.178)	(0.528)
U o .25 (1.15) u	0.13	0.50	1.00	0.88	0.75	1.50	1.00	2.75	2.25	1.88	2.119	97.753
	(0.24)	(0.36)	(0.68)	(0.75)	(0.14)	(0.29)	(0.36)	(0.66)	(0.78)	(0.65)	(0.093)	(0.645)

sequences. With a randomised multiple-session input sequence, at any point in the adaptive recognition process, the correct template will not, in general, correspond so well to the current input utterance as in the case of single-session input, since any previous utterances on which it has been adapted were not necessarily from the same session.

Secondly, in the cases in table 6.8, the numbers of occurrences of the different digits during the first 50-digit subsequence are likely to be unequal, so that the adaptation proceeds more unevenly than in the cases in table 6.7 (where each 50-digit sequence contains exactly five occurrences of each of the digits). Even with appropriate compensation, it may be expected that adaptation which proceeds unevenly across the vocabulary will not improve the overall recognition as much as adaptation which is more evenly spread.

In the first of these respects, the results in table 6.7 are more realistic than those in table 6.8: normally, in practical use of a recognition system, successive utterances will be spoken on the same occasion, with only occasional longer gaps in the chronological sequence. In the second respect, however, the results in table 6.8 are more realistic.

The results from the evaluation phase of the second series of experiments are set out in tables 6.9 and 6.10. The figures in table 6.9 are the means and standard errors (computed, as usual, from the average results for individual speakers) of the improvements in accuracy on the sequences of 200 evaluation utterances resulting from prior adaptation of the templates. Those in table 6.10 are the means and standard errors of the actual accuracies obtained on the evaluation data.

The compensation factors applied to the adapted templates' distances during the evaluation phase were the same as those applied (after the same

Table 6.9: improvements in digit recognition accuracy resulting from prior adaptation of templates (averaged over 10 template sets per speaker)

Adaptation and compensation	Mean (standard error) of improvement in percentage accuracy after adaptation												
	number of input utterances for adaptation												
	10	20	30	40	50	60	70	80	90	100	150	200	250
S o .25 -.05 u	0.78	1.26	1.43	1.90	2.14	2.22	2.42	2.52	2.54	2.65	2.72	2.77	2.79
	(0.09)	(0.15)	(0.10)	(0.21)	(0.19)	(0.22)	(0.24)	(0.26)	(0.29)	(0.32)	(0.35)	(0.40)	(0.44)
S o .5 -.05 h	0.56	1.09	1.43	1.84	2.04	2.36	2.51	2.60	2.64	2.71	2.79	2.82	2.84
	(0.14)	(0.26)	(0.18)	(0.25)	(0.18)	(0.20)	(0.21)	(0.24)	(0.26)	(0.27)	(0.31)	(0.35)	(0.39)
S o .5 -.05 j	0.48	1.16	1.46	1.88	2.10	2.39	2.53	2.60	2.67	2.73	2.79	2.82	2.84
	(0.15)	(0.23)	(0.18)	(0.27)	(0.20)	(0.23)	(0.23)	(0.24)	(0.27)	(0.28)	(0.31)	(0.35)	(0.40)
S o .5 -.05 (j) k	0.34	1.16	1.49	1.92	2.12	2.38	2.54	2.60	2.68	2.72	2.79	2.82	2.83
	(0.15)	(0.20)	(0.21)	(0.26)	(0.20)	(0.23)	(0.23)	(0.24)	(0.27)	(0.28)	(0.31)	(0.35)	(0.40)
S o .5 -.05 i	0.22	1.04	1.51	1.95	2.16	2.43	2.54	2.59	2.66	2.71	2.78	2.78	2.78
	(0.22)	(0.16)	(0.20)	(0.25)	(0.21)	(0.25)	(0.24)	(0.25)	(0.30)	(0.32)	(0.34)	(0.38)	(0.43)
U o .2 (1.15) u	0.52	0.86	1.03	1.36	1.63	1.68	1.95	2.07	2.18	2.23	2.36	2.39	2.39
	(0.13)	(0.13)	(0.06)	(0.10)	(0.04)	(0.13)	(0.12)	(0.06)	(0.12)	(0.11)	(0.17)	(0.20)	(0.24)
U o .25 (1.15) u	0.60	0.94	1.14	1.36	1.69	1.67	1.91	2.01	2.10	2.15	2.12	2.17	2.16
	(0.11)	(0.14)	(0.06)	(0.12)	(0.06)	(0.11)	(0.15)	(0.11)	(0.10)	(0.06)	(0.04)	(0.11)	(0.16)

Table 6.10: digit recognition accuracies with unadapted and adapted templates (averaged over 10 initial template sets per speaker)

Adaptation and compensation	Mean (standard error) of percentage accuracy after adaptation													
	number of input utterances for adaptation													
	0	10	20	30	40	50	60	70	80	90	100	150	200	250
S o .25 -.05 u	95.84	96.61	97.10	97.27	97.74	97.98	98.06	98.26	98.36	98.38	98.49	98.56	98.61	98.63
	(0.74)	(0.76)	(0.59)	(0.65)	(0.54)	(0.60)	(0.58)	(0.57)	(0.54)	(0.50)	(0.45)	(0.44)	(0.38)	(0.36)
S o .5 -.05 h	95.84	96.39	96.93	97.27	97.68	97.88	98.19	98.35	98.44	98.48	98.55	98.63	98.66	98.68
	(0.74)	(0.76)	(0.57)	(0.60)	(0.50)	(0.60)	(0.59)	(0.61)	(0.58)	(0.53)	(0.50)	(0.47)	(0.44)	(0.39)
S o .5 -.05 j	95.84	96.31	97.00	97.30	97.71	97.94	98.23	98.37	98.44	98.51	98.57	98.63	98.66	98.68
	(0.74)	(0.77)	(0.59)	(0.61)	(0.49)	(0.59)	(0.58)	(0.61)	(0.58)	(0.52)	(0.50)	(0.47)	(0.44)	(0.39)
S o .5 -.05 (j)k	95.84	96.18	97.00	97.33	97.76	97.96	98.22	98.38	98.44	98.52	98.56	98.63	98.66	98.67
	(0.74)	(0.79)	(0.60)	(0.59)	(0.49)	(0.59)	(0.59)	(0.61)	(0.58)	(0.51)	(0.50)	(0.47)	(0.44)	(0.39)
S o .5 -.05 i	95.84	96.06	96.88	97.35	97.79	98.00	98.27	98.38	98.43	98.49	98.54	98.61	98.62	98.62
	(0.74)	(0.80)	(0.60)	(0.59)	(0.51)	(0.60)	(0.56)	(0.57)	(0.56)	(0.48)	(0.47)	(0.44)	(0.42)	(0.38)
U o .2 (1.15) u	95.84	96.36	96.70	96.87	97.19	97.47	97.51	97.79	97.91	98.01	98.07	98.20	98.23	98.23
	(0.74)	(0.67)	(0.64)	(0.70)	(0.73)	(0.77)	(0.85)	(0.80)	(0.77)	(0.64)	(0.64)	(0.58)	(0.55)	(0.53)
U o .25 (1.15)u	95.84	96.44	96.78	96.98	97.19	97.52	97.51	97.75	97.84	97.94	97.99	97.96	98.01	98.00
	(0.74)	(0.70)	(0.62)	(0.70)	(0.68)	(0.73)	(0.79)	(0.82)	(0.81)	(0.70)	(0.70)	(0.71)	(0.66)	(0.64)

numbers of adaptations) during the adaptation phase, except for one case where compensation "j" was used in the adaptation phase but was replaced by compensation "k" in the evaluation phase. (This was done in order to compare the two sets of compensation factors without having to run the computationally intensive adaptation phase twice.)

It is clear from tables 6.9 and 6.10 that the recognition accuracy attained increases progressively as the templates are adapted. The average increase in accuracy after adaptation on 250 utterances was 2.84%, from 95.84% to 98.68%, in the best cases of supervised adaptation; or 2.39%, from 95.84% to 98.23%, in the better of the two unsupervised adaptation cases. In each case, more than half of this improvement was attained over the first 30 or 40 input utterances for adaptive recognition, and after 100 inputs the accuracy improved only slightly (by less than 0.2%) with continuing adaptation to the next 150 inputs. This suggests that, in general, if the size of the vocabulary to be recognised is V, and the words occur randomly with equal probability in the input sequence, and there is no significant "drift" (systematic change) in the speaker's voice and pronunciations over the period of use of the recogniser, near-optimal performance will be attained after adaptive recognition of about $10V$ utterances.

The improvement was greater and more rapid with supervised adaptation than with unsupervised, on the whole. The advantage of supervised adaptation over unsupervised was fairly, but not entirely, consistent across different speakers and amounts of input. The mean difference in the results after 250 inputs, between the supervised case with weight 0.5 and compensation "j" and the unsupervised case with weight 0.2 and compensation "u", was 0.44%, and the estimated standard error of this difference was 0.17, yielding a confidence of 0.96 for the superiority of the supervised adaptation; for the more directly comparable results with weight 0.25 and compensation "u", the result after supervised

adaptation was on average 0.63% better, and the standard error estimate was 0.31, yielding confidence 0.93.

For supervised adaptation, two weighting options were evaluated. The case with input weight 0.25 in the first adaptation corresponds to giving three times as much weight to the initial template as to each input utterance, whereas input weight 0.5 corresponds to equal weighting of all utterances (given that each initial template is derived from a single utterance). The latter case should produce the optimal estimates of the speaker's typical realisations of the words, after any specified amount of input. However, it can be seen from the results that the improvement attained after the first 10 inputs is greater with input weight 0.25, and it is only after larger amounts of input that the equal-weighting option begins to yield an advantage. The difference between the results after 10 inputs with the different weights is consistent across the four speakers: the mean difference is 0.22% (with compensation factors "h" and "u") or 0.30% ("j" and "u"), and the estimated standard error of this difference is 0.06 (in either case), giving confidence 0.98 or 0.99 respectively. The differences after larger amounts of adaptation input are smaller and less consistent. It should be noted that the results with input weight 0.25 may not be optimal for that weight value, since the compensation factors used were not optimised. With input weight 0.5, compensation "j" appears to be nearly optimal over long input sequences, though the results with compensation "h" or "k" are very similar, especially after 20 or more input utterances.

For unsupervised adaptation, initial input weight values of 0.2 and 0.25 were tested. After up to 50 input utterances, the larger weight yielded a greater improvement, but after longer input sequences the results with the smaller input weight were better on average. The danger in having a large input weight in unsupervised adaptation is that adaptation to a misrecognised input

may seriously corrupt a template, and result in repeated occurrences of the same misrecognition, and hence repeated incorrect adaptations. With the smaller weight on the input, the total number of incorrect adaptations occurring (for all speakers, over the 450 inputs in each trial: 36000 recognitions in all) was 256; with the larger weight, it was 303. The main component of the difference between these numbers came from one instance of instability, for the third template set for speaker 3, where "0" was repeatedly recognised as "7" and hence the "7" template was repeatedly adapted to utterances of "0". (With the smaller input weight, this happened with only one of the two input sequence orderings, but with the larger weight it occurred with both of them.) The statistical significance of the comparison of weight values on the basis of these experiments is moderate (the mean difference in accuracy after 250 inputs was 0.23%, and its estimated standard error was 0.11, yielding confidence 0.94); but it seems reasonable to suppose that there will be some initial input weight value (which may be 0 or some positive number) above which the risk of instability increases.

As in the previous series of experiments (section 6.3.2.4), the improvements attained through adaptation were greatest for the speaker (speaker 3) who had the poorest recognition accuracy without adaptation. This is revealed in the standard error figures in tables 6.9 and 6.10, especially those for recognition after supervised adaptation. As the adaptation progresses, the variability across speakers in the actual percentage recognition accuracy (table 6.10) is reduced, whereas the variability in the improvement due to adaptation (table 6.9) increases because the improvement for the worst-recognised speaker becomes increasingly greater than the improvements for the other speakers. This tendency for the individual speakers' results to be pulled closer together can be seen also in figures 6.7 and 6.8, where results with supervised adaptation and with unsupervised adaptation are plotted against numbers of input utterances. The

Figure 6.7: results after supervised adaptation on up to 250 digits

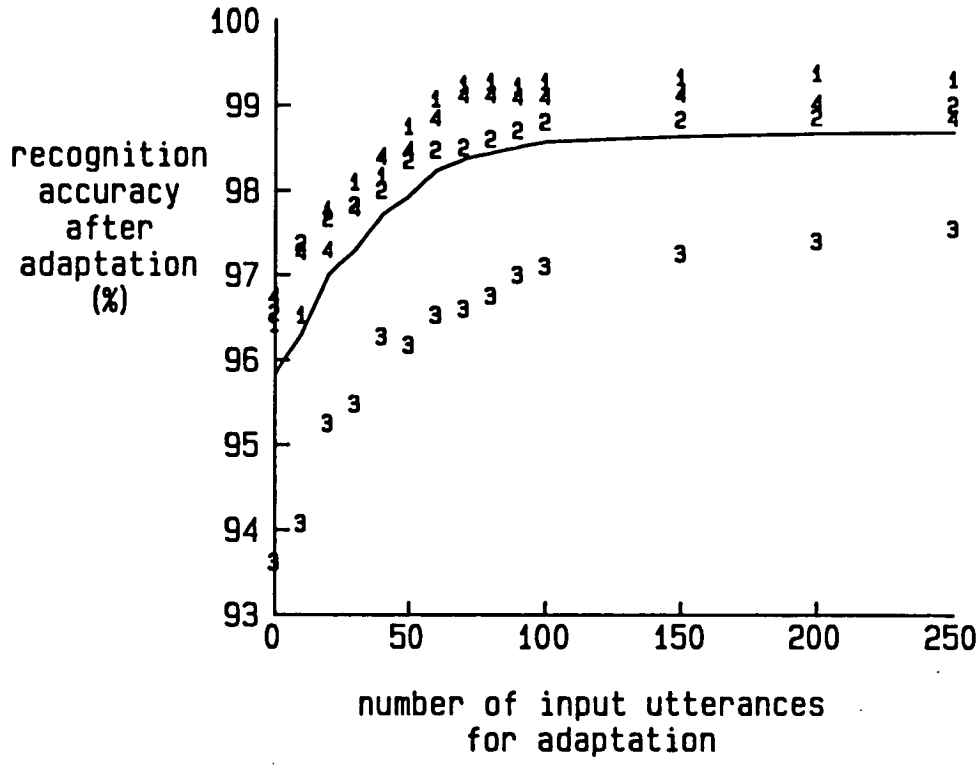
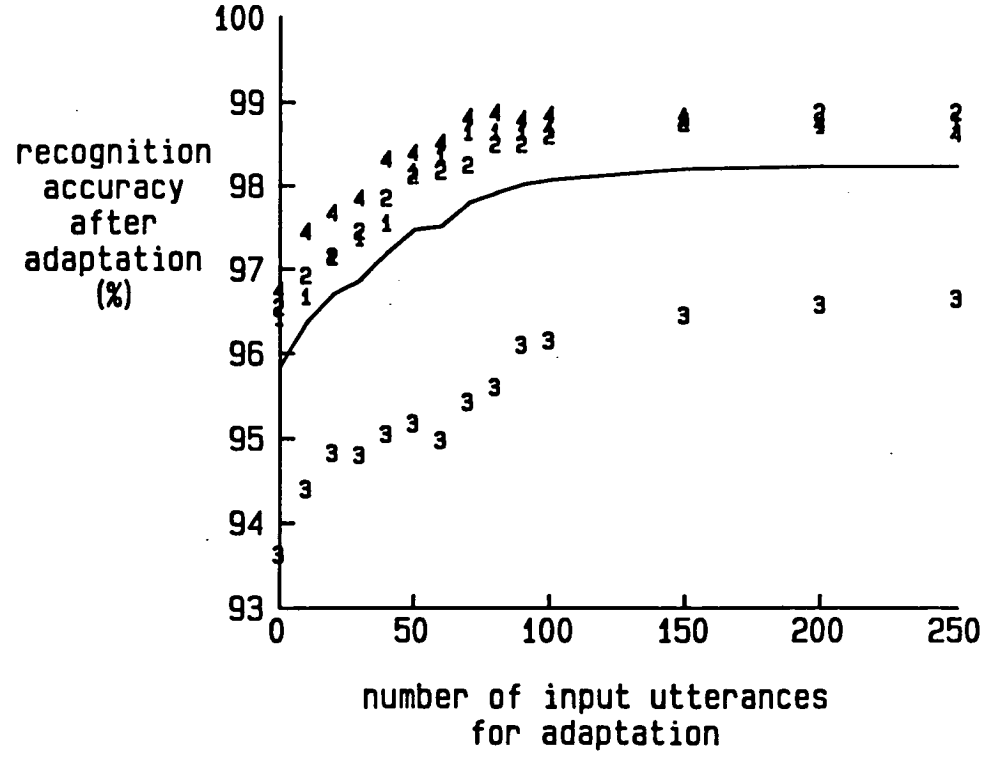


Figure 6.8: results after unsupervised adaptation on up to 250 digits



factors by which the speakers' recognition error rates were reduced after supervised adaptation (on 250 inputs) ranged from 2.6 (speaker 3: 6.38% to 2.45%) to 5.1 (speaker 1: 3.58% to 0.70%); after unsupervised adaptation, from 1.9 (speaker 3: 6.38% to 3.35%) to 3.1 (speaker 2: 3.45% to 1.12%). (It can be seen from these results that, although the absolute difference in recognition accuracy, or equivalently in error rate, due to adaptation was greatest for speaker 3, whose results with unadapted templates were poorest, the proportional reduction in error rate for this speaker was the smallest.)

As well as improving the accuracy of recognition, adaptation of the templates reduces the numbers of templates matched against the input at the second and third stages of the recognition procedure, and thus improves the speed of recognition. The numbers of template matches per recognition at the second and third stages (in the evaluation phase of the experiments) are plotted against the number of utterances for adaptation, for the best cases of supervised and unsupervised adaptation, in figures 6.9 and 6.10. (The results plotted are for only one of the two random orders of input, for each template set; but the effects of adaptation are quite consistent across different speakers and template sets, and should likewise be consistent across different input orders for the same template sets.) The saving in computation due to this effect of adaptation is still less than the extra computation required for the adaptation itself. Thus, while adaptive recognition should become slightly faster as it proceeds, it will still take more computation than non-adaptive recognition, unless the adaptation is switched off after a while (as it was between the adaptation and evaluation phases of these experiments), or unless the faster linear alignment option is used in the adaptation.

Figure 6.9: second and third stage matching statistics with supervised adaptation

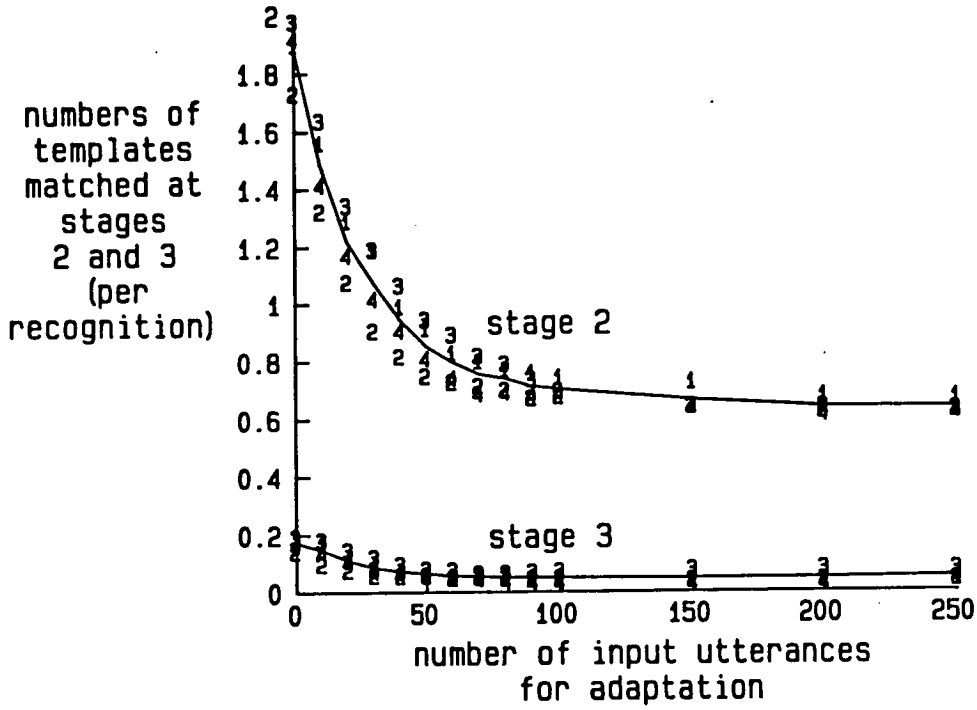
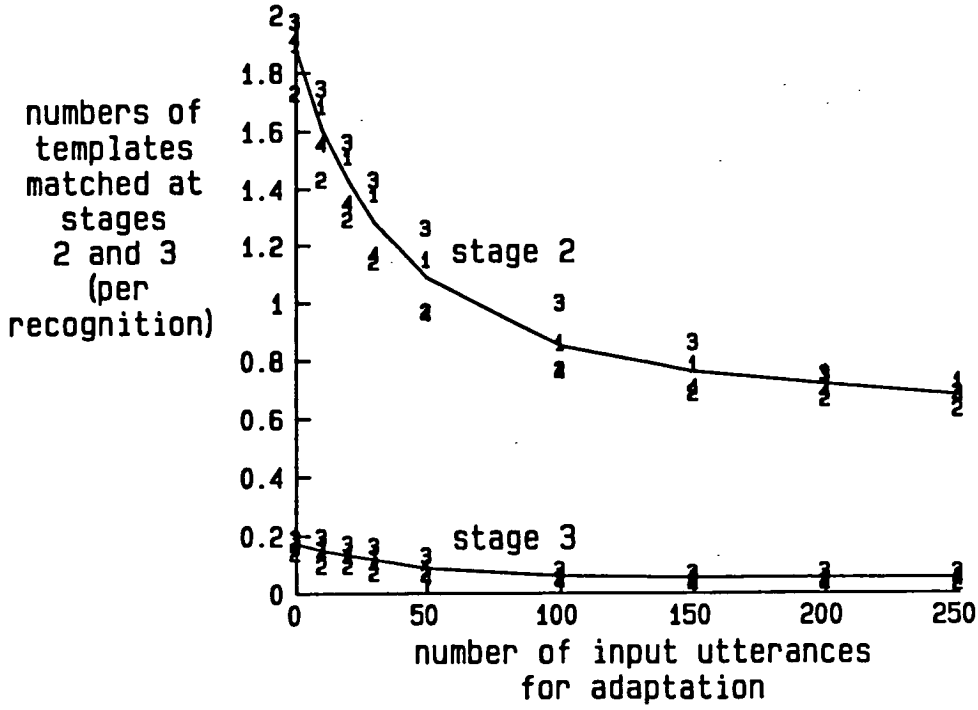


Figure 6.10: second and third stage matching statistics with unsupervised adaptation



6.4: Discussion of speaker-specific template adaptation results

Some comments on specific results have already been given, alongside the results, in section 6.3.2. The following remarks summarise the findings of the experiments in more general terms.

Firstly, it is clear from the results that adaptation can substantially improve speaker-specific templates which are initially derived from one or two utterances of each word. After several adaptations of each template, using the optimisation weighting, the recognition error rate with the adapted templates is typically about half what it would be with the unadapted templates, if each of these is derived from a single training utterance. In some cases considerably more than half of the errors are eliminated (e.g. for speakers 1, 2 and 4 in the digit recognition experiments). Smaller improvements have been observed for two-token initial templates (as shown in tables 6.2 and 6.5). The exact degree of improvement attained will depend on many variables, such as the incidence of confusable word pairs in the vocabulary, the representativeness of the initial training data, the consistency of the speaker and the variability of background noise. Greater improvements can be achieved when the adaptation is supervised, since stability can be ensured without the imposition of a threshold condition which restricts the occurrences of adaptation. However, substantial improvements have been demonstrated with unsupervised adaptation. When the adaptation is unsupervised, instabilities are liable to occur occasionally, in which particular misrecognitions and wrong adaptations occur repeatedly; to correct this, a retraining facility should be provided.

Secondly, for the attainment of the optimal improvement with adaptation it is important that appropriate compensation factors be applied to the word distances, to allow for the tendency for adapted templates' distances to be smaller

than those for unadapted templates. The values of the optimal compensation factors will depend on the adaptation weighting. In the above experiments (using optimisation weighting), the best results with supervised adaptation were obtained with initial input weight 0.5 (equal weighting of all training and adaptation data) and the compensation factors identified (in table 6.1) as "j"; the best results with unsupervised adaptation were with initial input weight 0.2 and compensation "f".

Where the adaptation is supervised, negative adaptation to misrecognised inputs, with a small negative weight on the input, appears to yield a slight enhancement of accuracy; from the experiments conducted, the statistical significance of this result is moderate.

Two basic types of weighting were formulated for use in template adaptation: the tracking form and the optimisation form. Because of the experimental design using randomly permuted input sequences, only the optimisation form has been thoroughly evaluated in these experiments. Some limited experiments were conducted with tracking adaptation (without reordering of the input utterances). The results (in tables 6.2, 6.5 and 6.6) show that tracking adaptation, with input weight 0.2, can yield recognition improvements similar to those with optimisation (initial input weight 0.2 or 0.25), or slightly better, over five to 20 repetitions of the vocabulary; but the statistical significance of the comparison between tracking and optimisation is low. A full comparison of optimisation and tracking would require the use of a large amount of data (to ensure statistical significance, given the limitations imposed by the need to retain the chronological order), and separate determination of the optimal initial input weight values and compensation factors for the two forms of weighting. Moreover, the results would be affected by the lengths of the data collection sessions, and the lengths of time between successive sessions – though variations in these quantities

could be simulated by collecting the utterances in long sessions close together in time, and then selecting sessions or parts of sessions for use in the experiments.

It may be expected that some form of adaptation which takes account of gradual drift in time will yield the best possible results – better than those attained with the optimisation form which takes no account of this phenomenon. However, the tracking form of weighting, as currently formulated, is probably not optimal, since it treats all templates the same, regardless of whether they are previously adapted or unadapted. It seems reasonable that unadapted templates (formed on a previous occasion) will be the least representative of the realisations of the words in the current recognition session; templates adapted during a previous session will be somewhat more representative, as will templates formed during the current session but not yet adapted; and templates adapted several times during the current session will be the most representative of all. The more reliable a template, the more it should be weighted in adaptation (or, equivalently, the less each new input used in adapting it should be weighted). A possible method of adaptation, allowing for all these factors, is as follows. During a recognition session immediately following the initial formation of a template, begin by using the optimisation weighting, with initial input weight w_{new}, and then change to the tracking form of weighting, with input weight w_{track}, after n_{new} adaptations. (Here w_{new} should be fairly large – perhaps 0.5 in the case of single-utterance training and supervised adaptation. For a smooth transition from optimisation to tracking, w_{track} should be between the input weight values, derived by the optimisation method, for the n_{new}th and $(n_{new}+1)$th adaptations, i.e.

$$\frac{w_{new}}{1+(n_{new}-1)w_{new}} \geq w_{track} \geq \frac{w_{new}}{1+n_{new}w_{new}} \qquad (6.1)$$

(using equations (5.1), (5.2) and (5.4)).) During a subsequent recognition session, begin with the optimisation weighting again, with initial input weight $w_{\text{old},k}$ for a template previously adapted k times, and change to tracking (with input weight w_{track} again) after $n_{\text{old},k}$ adaptations, where

$$\frac{w_{\text{old},k}}{1+(n_{\text{old},k}-1)w_{\text{old},k}} \geq w_{\text{track}} \geq \frac{w_{\text{old},k}}{1+n_{\text{old},k}w_{\text{old},k}}. \tag{6.2}$$

(The size of $w_{\text{old},k}$ should decrease as k increases; also, $w_{\text{old},0}$ should be greater than w_{new}, and $w_{\text{old},k}$ should be greater than w_{track} for all values of k. A possible formula for $w_{\text{old},k}$ is

$$w_{\text{old},k} = \frac{1}{v_{\text{old},0}+ka+1} \tag{6.3}$$

for $k \leq n_{\text{new}}$, and

$$w_{\text{old},k} = \frac{1}{v_{\text{old},0}+A+1} \tag{6.4}$$

for $k > n_{\text{new}}$ – where the relative weight $v_{\text{old},k} = v_{\text{old},0}+ka$ is a scaled-down version ($a < 1.0$) of the first-session relative weight $v_{\text{new},k}$ computed according to (5.1), and $n_{\text{new}}a \leq A \leq (n_{\text{new}}+1)a$.) With this system of weighting, which distinguishes between previous-session and current-session adaptations, a correspondingly more sophisticated system of compensation factors would be required. A problem with such a complex form of adaptation is that the number of parameters to be specified is considerably larger than with either of the two basic forms already described. Estimating optimal values of these parameters would require extensive experiments.

6.5: Observations on the interactive recognition sessions

Statistics of the interactive recognition sessions in which the four-speaker digits data base was collected are set out in table 6.11. Under "date and time", the format is "year month day hour minute", where each component is expressed as a two-digit number. (The time given is for the start of the session.) In the next column is a template code followed by an algorithm parameter code and a compensation code. The template code is "1" for single-token templates and "2" for two-token averaged templates; in most cases, the templates were adapted ones (usually the results of the preceding session's adaptation), but unadapted templates, where these were used, are marked by "#" (if created on a previous occasion) or "*" (if created on the current occasion). The algorithm parameter codes are as follows:-

A: $r_2 = 1.05$; $r_3 = 1.15$; endpoint adjustment;

B: $r_2 = 1.05$; $r_3 = 1.15$;

C: $r_2 = 1.01$; $r_3 = 1.05$.

In each case, the template elimination thresholds were $t_1 = 1.6$ and $t_2 = 1.2$; the threshold on the word distance ratio for adaptation was equal to the rejection threshold r_3; the adaptation operated in the supervised mode using "CORRECTION" (as described in section 5.3); the tracking form of adaptation weighting was employed, with input weight 0.2; and there was no negative adaptation. The compensation codes are as in table 6.1, with "0" for "no compensation". The compensation factors applied were based on the numbers of adaptations so far within the current session: there was no use of stored adaptation counts for previously adapted templates. The statistics of each session are given in the next six columns: these contain, respectively, the total number of utterances in the session; the numbers of correct recognitions, errors (wrong recognitions) and

Table 6.11: statistics of interactive digit recognition sessions used to collect the four-speaker data base

Speaker (sex)	Date and time	Templates and system parameters	Numbers of words						Retrain	Comments
			total	cor	err	rej	cor+	err+		
1 (m)	8706171216	1# A 0	47	40	1	6	44	3	9	24CC; 0-STOP
	8706171247	1 A 0	18	15	0	3	16	2	STOP	24CC
	8706171329	1 A 0	31	29	0	2	30	1		24CC; NS
	8706171352	1 A 0	23	22	0	1	23	0		24CC
	8706261402	1# A 1	93	91	0	2	93	0		
	8707031713	1 B a	104	101	0	3	103	1		
	8707171302	2 B a	54	53	0	1	54	0		
	8707220916	2 B a	53	53	0	0	53	0		
	8708031706	2 B c	51	51	0	0	51	0		
	8708040849	2 B c	51	51	0	0	51	0		
	8708060855	2 B c	51	51	0	0	51	0		
2 (m)	8707131400	2* B b	52	51	0	1	52	0		
	8707141635	2 B 0	52	51	0	1	52	0		
	8707171621	2 B a	53	51	0	2	51	2		
	8707201701	2 B a	52	51	0	1	52	0		
	8707221704	2 C a	44	43	1	0	43	1		6-STOP
	8707221722	2 B a	10	9	0	1	10	0	STOP	
	8707240926	2 B a	54	51	0	3	52	2		
	8707271518	2 B a	52	51	0	1	52	0		
	8707301444	2 B a	53	51	0	2	53	0		
	8707311259	2 B c	54	51	0	3	52	2		
	8708201005	2 B c	53	51	0	2	52	1		
3 (f)	8707141401	2# B e	53	52	1	0	52	1		3-8
	8707171558	2 B a	53	51	0	2	51	2		
	8707211427	2 B a	69	57	4	8	61	8		0-5,3-8,1-5,STOP-7
	8707221352	2 C n	55	54	1	0	54	1		1-5
	8707241321	2 B a	51	51	0	0	51	0		
	8707270905	2 B a	52	51	0	1	51	1		
	8708041004	2 B c	56	53	2	1	54	2		9-5,4-5
	8708050917	2 B c	54	51	0	3	53	1		
	8708061222	2 B c	55	52	1	2	52	3		4-5
	8708201332	2 B c	53	51	0	2	53	0		
4 (f)	8707150908	2* B n	51	51	0	0	51	0		
	8707171234	2 B a	52	51	0	1	52	0		
	8707220856	2 B a	58	53	0	5	56	2		
	8707230924	2 B a	52	51	0	1	52	0		
	8707241245	2 B a	55	53	0	2	55	0		
	8707281320	2 B a	51	51	0	0	51	0		
	8707311318	2 B c	51	51	0	0	51	0		
	8708271417	2 B c	58	54	1	3	56	2		STOP-7
	8709091401	2 B c	55	54	1	0	54	1		1-7
	8709100927	2 B c	52	51	0	1	52	0		

rejections; and the numbers of correct recognitions and errors obtained with the rejection thresholds set to 1.0 to force a recognition of every input. Under "Retrain" are listed any words whose templates were retrained during the sessions. (In each case, the number of utterances used in retraining was the same as the number of training utterances used to form the initial template.) Under "Comments", "24CC" means that the full set of 24 cepstral coefficients per frame was used, instead of only the first 12; an entry of the form "word1-word2" means "word1 was misrecognised as word2"; and "NS" means that the adapted templates were not saved for use in the next recognition session.

In each case listed in table 6.11, the input sequence consisted of digits in the standard order (as listed in section 6.2.4), beginning where the previous session had ended if the 50-digit sequence had not been completed in the previous session, and finishing with "STOP" unless some other word was misrecognised as "STOP". Additions to the standard sequence of words resulted from misrecognitions and rejections (after which the speaker said "CORRECTION", in case of misrecognition, and repeated the word which had not been recognised); human errors (where the speaker said the wrong word -- followed by "CORRECTION"); occurrences of "RETRAIN"; and, in speaker 1's fifth session, the inclusion of four repetitions of the 10 digits in numerical order, to provide data for the initial estimation of compensation factors. There were eight instances of "human error"; in three of these cases, the error consisted of going on to the next word in the sequence when the word preceding it had not been recognised.

The sessions listed in table 6.11 are those which were actually used to provide utterances for the data base. In addition to these, during the period of the data base collection, there were several digit recognition sessions which were not used -- in some cases because the input data were accidentally lost, and in other cases because only part of the 50-digit sequence was collected in a session, and it

was considered preferable to take each set of 50 digits from a single session (or, failing that, from sessions with only a few minutes between them). There were also some digit recognition sessions (with speaker 1 only) in which more irregular sequences of digits were spoken instead of the standard 50-digit sequence. (These were used to provide the data sets d1a, d1b and d2 for the compensation experiments described in section 6.3.2.1.) These additional digit recognition sessions are listed in table 6.12. The notation is the same as in table 6.11, with the addition that "D" under "Templates and system parameters" is the same as "A"

Table 6.12: statistics of additional digit recognition sessions

Speaker (sex)	Date and time	Templates and system parameters	Numbers of words						Retrain	Comments
			total	cor	err	rej	cor+	err+		
1 (m)	8706161618	1# A 0	27	22	0	5	24	3		24CC
	8706171148	1# A 0	12	7	0	5	11	1		24CC
	8706261356	1 A 1	14	12	1	1	13	1		2-STOP; NS
	8707011717	1 D 0	16	14	1	1	15	1		7-STOP; NS
	8707011723	1 D 0	4	3	1	0	3	1		7-STOP; NS
	8707011727	1 D 0	38	38	0	0	38	0	7,STOP	NS
	8707021240	1* A u	174	163	0	11	170	4		
	8707031034	1* B n	174	161	1	12	168	6	RETRAIN	3-RETRAIN
	8707061725	1* B n	87	76	3	8	79	8	7	7-1,7-6,6-1
	8707081231	2* B n	213	203	1	9	208	5	RETRAIN	8-RETRAIN
	8707150927	2 B a	216	209	0	7	214	2		
	8707151328	2 B a	23	21	0	2	22	1		
	8707151716	2 B a	31	31	0	0	31	0		
2 (m)	8707091232	1* B 0	32	29	3	0	29	3	9	9-5,4-5,2-STOP
	8707091244	1 B 0	49	32	4	13	39	10	STOP,STOP, 7,STOP	STOP-7 (x4)
	8707311254	2 B c	6	5	1	0	5	1		7-STOP; NS
3 (f)	8707091315	1* B 0	67	54	2	11	58	9	0	0-7,0-5
	8707131443	2* B b	41	38	0	3	40	1		NS
4 (f)	8707131316	1* B b	56	53	1	2	54	2		STOP-7

without the adaptation. The 160-digit sequences d1a and d1b were taken from the sessions dated 8707021240 and 8707031034 respectively, and the 200-digit sequence d2 was taken from the session dated 8707150927.

Accumulating all the results in tables 6.11 and 6.12, the total number of input utterances is 3471, of which 3282 (94.6%) were correctly recognised, 32 (0.9%) misrecognised and 157 (4.5%) rejected. Without the elimination option, 3373 (97.2%) would have been correctly recognised, and 98 (2.8%) misrecognised.

Counting only the 2000 utterances which were included in the data base, 1939 (96.95%) were correctly recognised; 9 (0.45%) were misrecognised; 52 (2.6%) were rejected; and, without the rejection option, there would have been 1969 (98.45%) correctly recognised and 31 (1.55%) misrecognised. This "without rejection" accuracy is higher than was attained in the main series of experiments with the data base, even with optimal adaptation and compensation; but those experiments used single-token initial templates, did not allow retraining in cases of persistent error, and took no account of the chronological sequence of the utterances. With optimal adaptation and compensation (as in the main experiments), and appropriate retraining (as in the interactive sessions), higher accuracies should be possible than were demonstrated in sections 6.3.2.4 and 6.3.2.5.

Of the 32 recognition errors, 14 involved the word "STOP". Two reasons for this may be suggested: the difficulty of endpoint detection (during training or recognition) for a word containing two stop consonants, and the less frequent occurrence (and hence adaptation) of "STOP" in the input sequences. The latter should have been countered to some extent by the use of compensation factors; but the compensation applied during the interactive sessions was not optimal in that no record of the number of adaptations of each template was kept from one session to the next, and so all the templates were assigned the same

compensation factor (1.0) at the beginning of each session even though the control words' templates might be considerably less adapted than the others. (The problem of compensating for previous-session and current-session adaptation is rather complex: adaptation during a previous session will tend to reduce a template's distances in the current session, but not as much as adaptation during the current session does. A refinement of the compensation technique – with or without a refinement of the adaptation weighting as described in section 6.4 – may be required to treat this problem adequately.) The other two control words caused less difficulty: there were no errors involving "CORRECTION" (though it was sometimes rejected), and only two involving "RETRAIN".

Retraining was invoked 14 times, including twice by accident when digits were recognised as "RETRAIN". It was mainly with single-token initial templates that retraining was found necessary.

Details of interactive sessions conducted using the 53-word "W" vocabulary are listed in table 6.13. These sessions were conducted for only one speaker. The vocabulary is larger than the 13-word "F" vocabulary (digits plus control words) used for the sessions in tables 6.11 and 6.12, and is considerably more difficult for recognition, since it contains pairs of words like {seventeen,seventy} and {thirty,Thursday}; this difficulty is reflected in the greater frequencies of errors, rejections and retrainings occurring. The notation in table 6.13 is the same as in tables 6.11 and 6.12. In the "Retrain" and "Comments" columns, the number words are expressed in figures, and the day and month names are abbreviated, to save space.

The totals of the results in table 6.13 are 3362 utterances; 2708 (80.5%) correct; 63 (1.9%) misrecognised; 591 (17.6%) rejected; and, with no rejection option, 2926 (87.0%) correctly and 436 (13.0%) wrongly recognised. There is a

Table 6.13: statistics of interactive recognition sessions with the "W" vocabulary (speaker 1)

Date and time	Templates and system parameters	Numbers of words						Retrain	Comments
		total	cor	err	rej	cor+	err+		
8706181338	1* A 0	88	63	5	20	70	18	1000,18,19, Thu,Thu,40,40	18-15,Thu-30(x4)
8706221232	1 A 0	136	110	6	20	117	19	70,Thu,40	Thu-30(x4),40-14,40-30
8706241336	1# A 0	89	58	4	27	68	21	Thu,Sun,16	8-May,Sep-6,16-6,80-18
8706261259	1 A 1	46	33	1	12	39	7		Tue-Thu; NS
8706261320	1 A 1	56	48	0	8	49	7	7,Wed	NS
8706261343	1 A 1	35	28	0	7	28	7	50,Wed	
8706291550	1 A a	151	116	2	33	129	22	Tue(x3),10	Tue-Thu,40-4
8706301209	1* A n	496	346	10	140	392	104	19(x2),1,40,30, Thu(x5),Wed,14, 20(x2),13,Jun(x2), Tue,Dec,50(x3)	6-60,13-3,13-10,19-13, 30-40,50-15(x3), Wed-Mon,Wed-Aug
8707100923	2# B u	123	104	3	16	114	9		17-70,30-Thu,60-6
8707101634	2 B a	123	103	1	19	115	8	Apr	70-7
8707111636	2 B a	344	313	5	26	325	19	Wed	60-6(x2),Apr-8, 30-Thu,Thu-30
8707161713	2 B a	115	101	1	13	106	9		15-50
8707171654	2 B a	112	102	0	10	107	5	90	
8707240948	2 B a	36	26	0	10	32	4	70	
8707271628	2 B a	242	199	6	37	207	35		70-7,40-14,80-8, 16-60,50-15(x2)
8707281123	2 B a	361	310	7	44	323	38		Apr-8,80-18,19-90(x2), 30-Thu,14-40,40-4
8707291044	2 B a	366	308	4	54	322	44	50	17-70(x2),14-40,90-19
8707291705	2 B a	151	121	4	26	133	18		14-40,13-30,Apr-8
8707301509	2 B a	83	56	1	26	72	11	90,70	4-Oct
8707301700	2 B a	76	53	1	22	57	19	Thu	Wed-70
8708041039	2 B a	64	54	1	9	58	6		50-60
8708071215	2 B a	69	56	1	12	63	6		70-17

noticeable difference between the results for sessions starting with unadapted single-token templates ($\frac{530}{673}$, or 78.8% correct, with no rejections) and the results for sessions starting with adapted single-token templates ($\frac{362}{424}$, or 85.4%) or with two-token templates ($\frac{2034}{2265}$, or 89.8%). Among the results with

two-token templates, there is no discernible improvement with adaptation: indeed, the accuracy attained in the first session, with no prior adaptation, is $\frac{114}{123}$, or 92.7%, which is higher than the overall accuracy in the subsequent sessions ($\frac{1920}{2142}$, or 89.6%) – although this may not be significant, especially as the former result is derived from only one session and there were differences in the word order and compensation factors from session to session.

The single-token and two-token template sets used in the "W" vocabulary recognition sessions were the same (apart from the inclusion or omission of the control words) as those later used for the experiments described in sections 6.3.2.1 to 6.3.2.3. The input data sets used in those experiments were derived as follows: t1, from session 8706301209; t2, from 8707111636; t3, from 8707271628, 8707281123, 8707291044, 8707291705, 8707301509 and 8707301700; and t4, from the first six sessions (8706181338 to 8707261343).

The sessions for speaker 1 listed in tables 6.12 and 6.13 included some sessions considerably longer than those (of just over 50 words) used in the collection of the main digits data base. The longest session was that dated 8706301209, with the "W" vocabulary (table 6.13), which included 496 input utterances and took two hours and 42 minutes. Some of the other "W" recognition sessions also lasted over an hour. The time per recognition was typically slightly longer for this vocabulary than for the digits, both because most of the words were longer (so that the LPC analysis took longer) and because of the larger numbers of templates to be matched, especially at the second and third stages, due to the greater size and confusability of the vocabulary. Also, when the initial templates were formed from single utterances, retraining was required considerably more often for the "W" vocabulary than for the "F" vocabulary, because of persistent errors and rejections on certain words (such as "Thursday", "forty" and

"fifty") which were very similar to other words of the vocabulary. The "W" recognition sessions were thus more tedious and stressful for the speaker than the "F" sessions. The training session to construct the set of two-token "W" templates was also found rather long. For these reasons, no attempt was made to construct templates or conduct recognition sessions for the "W" vocabulary with speakers 2, 3 and 4.

CHAPTER 7

ADAPTATION OF SPEAKER-INDEPENDENT TEMPLATES

7: ADAPTATION OF SPEAKER-INDEPENDENT TEMPLATES

7.1: Introduction

The experiments reported in chapter 6 have shown the benefits of template adaptation in an initially speaker-trained word recognition system. It has been shown that adaptation, supervised or unsupervised, with appropriate compensation, can substantially improve the recognition performance of speaker-specific templates by incorporating more utterances (by the same speaker) into each template. It may be expected that even greater improvements will be attained if adaptation can be successfully applied to an initially speaker-independent template set. In this case, the initial unadapted templates incorporate no data from the speaker whose inputs are to be recognised (being formed instead from utterances by a standard set of training speakers); by adaptation, it should be possible to "tune" these templates to the current input speaker, improving their correspondence to that speaker's typical realisations of the words, so that they yield better recognition performance for the speaker concerned.

This chapter contains a description and results of experiments conducted with a 100-speaker digits data base, in which utterances from 50 of the speakers were used to form the initial speaker-independent templates, and utterances from each of the remaining speakers were used as input for adaptive and non-adaptive recognition. Section 7.2 gives a description of the data and initial template formation; section 7.3 records the experiments and results, and includes some analysis and discussion of details of the results; and section 7.4 contains a more general discussion.

7.2: Data base and template formation

7.2.1: The 100-speaker digits data base

A multiple-speaker data base had already been collected for use in another project at the Centre for Speech Technology Research (CSTR). This included recordings of the digits spoken three times by each speaker, as well as various other words and sentences. The digits uttered by 100 of the speakers were chosen for use in the speaker-independent template adaptation experiments. A brief account of the speaker set, data collection and processing is given below.

The set of 100 speakers consisted of 30 from Edinburgh (15 male and 15 female: mostly members of CSTR and their wives and husbands), and 70 (59 male and 11 female) from industrial and commercial sites in Maidenhead, Portsmouth and London. Thus there were 74 male and 26 female speakers in all. These were divided into two sets: 50 training speakers, and 50 test speakers — with 37 male and 13 female in each set. The two sets were made as similar as possible in their composition, in the sense of both having equal or nearly equal numbers of speakers of each possible combination of location and sex. One of the Edinburgh female speakers was subsequently excluded from the test set because of an error during the processing of the data, leaving a set of 49 test speakers (37 male and 12 female).

Each speaker participated in a single data collection session, in which three repetitions of the 10 digits, in random orders, were recorded. Each 10-digit sequence was recorded during a 30-second period, in which each digit in turn was displayed on a computer screen (printed in letters, to ensure the use of the pronunciation "zero" for "0") to prompt the subject to speak it. There was a short pause (until the speaker gave a "ready" signal by pressing a joystick button) between successive 10-digit sequences. The recordings were monitored by a

member of the research project, who initiated repetitions of any 10-digit sequences badly affected by noise or mispronunciation. The utterances were collected using a table-mounted Sennheiser MKH406 microphone, passed through a PCM digital coder, and recorded onto video cassette. The level of background noise varied among sessions, although efforts were made to reduce it by removing noise sources (when possible) and using a sound-absorbing screen around the speaker.

The utterances were transferred onto the MC550 computer using an analogue connection into the analogue-to-digital convertor. As with the data base of chapter 6, the sampling rate was 20kHz, with 12-bit resolution. Endpoint detection was performed by the method described in section 5.1 (but with different parameter settings from those listed in table 5.1). The detected digits in each sequence of 10 were rearranged into numerically increasing order. A 12th-order LPC analysis was applied, in a 25.6ms Hamming-windowed frame every 10ms, to derive 12 cepstral coefficients.

7.2.2: Formation of speaker-independent templates

The first utterances of all the digits by each of the 50 training speakers were taken as training data for the formation of speaker-independent templates. (The second and third repetitions of the digits by the training speakers were not used.)

A program was developed to perform clustering and template formation using the criterion based exchange (CEx) algorithm [100]. This clustering algorithm was chosen because it had been reported [100] to yield better template sets for speaker-independent recognition than the commonly-used K-means algorithm, and because it could operate, without much further computation, from a

precompiled table of distances among the training tokens of each word. The required distance tables were computed by the same program which had earlier been used to provide distances for multiple-stage recognition simulations (as mentioned in section 4.4.1).

For each word of the vocabulary, the clustering algorithm starts by reading in the table of word distances and selecting c cluster centres $C(0), \cdots, C(c-1)$, where c is the designated number of clusters per word of the vocabulary. (The cluster centres are used only in the initialisation of the clustering, and not in the subsequent criterion based exchange procedure.) The first cluster centre $C(0)$ is the first token in the (arbitrarily ordered) set of training tokens for the current word. Thereafter, each successive cluster's centre is chosen from among the training tokens so as to maximise the minimum distance between it and a previously chosen cluster centre. That is,

$$C(r) = \underset{n}{\operatorname{argmax}} \left(\min_{i<r} D(C(i),n)\right), \tag{7.1}$$

where D is the word distance function, and the integer n is used to identify the nth training token. Once the cluster centres have been determined in this way, all the training tokens are classified into clusters by a minimum-distance rule: i.e. the nth token is assigned to cluster i where i minimises $D(C(i),n)$.

It is possible to specify an exclusion parameter e, so that centres which result in clusters of e or fewer tokens are replaced until every cluster has more than e tokens in it. In this case, the process of determining new cluster centres and then classifying all the training tokens is iterated. At each iteration, the tokens which are centres of small clusters (if any) are excluded from consideration as possible cluster centres, so that the minimisation in (7.1) is carried out only over values of n which have not been excluded. (A token which is excluded

at a given iteration remains excluded at all following iterations, so that it can never again be tried as a cluster centre. Otherwise there would be the danger of an infinitely repeating loop.) The number of r values for which (7.1) is evaluated in each iteration (after the first) is equal to the number of cluster centres excluded at the previous iteration. (When (7.1) is to be applied in an iteration other than the first, the k cluster centres retained from the preceding iteration are first renumbered so that their numbering is consecutive from 0 to $k-1$; then (7.1) is evaluated for r from k to $c-1$.)

Once the initial clusters have been determined and, if necessary, adjusted by the small-cluster exclusion procedure, the criterion based exchange process begins. The criterion function is of the form

$$F = \sum_{i=0}^{c-1} \left(\min_{n \text{ in cluster } i} \sum_{m \text{ in cluster } i} D(n,m) \right) \qquad (7.2)$$

– that is, it is the sum of intra-cluster distances taken from cluster miniav centres. This quantity F is to be minimised by exchanging tokens among the clusters. In each iteration of the process, each token n is considered in turn, and if F can be reduced by moving n from its present cluster to a different one then it is moved (in such a way as to cause the greatest possible reduction in F). The procedure is iterated until no further reduction in F can be achieved by any single-token exchange operation. (Thus it always finds a local minimum of F, though this is not guaranteed to be the global minimum.)

A postprocessing step may be executed once the exchange process has terminated. In this case, any single-token clusters are augmented to size 2, by moving into each such cluster the nearest token which is in a cluster of more than two tokens. (This works, and takes k moves, where k is the number of single-token clusters, provided that the number of clusters c is not more than

half the number of training tokens for this word of the vocabulary; if c is too large, at least one single-token cluster must remain, since there are not enough tokens for two per cluster.)

When the partitioning of the training tokens into clusters has been completed, the tokens in the cluster are averaged together sequentially, using either DTW or linear averaging. (The averaging and weights are the same as in the optimisation form of adaptation, as described in section 5.3. With the DTW method, if a token cannot be averaged into the cluster template because it is too long or too short, it is not used.)

A length normalisation of each training token to 30 vectors, by linear segmentation and interpolation (as in the third stage of the recognition procedure), was applied in some cases before the averaging. This reduced the computation required for the averaging (since the average word length before normalisation was more than 30 frames), and ensured that no tokens were excluded because of excessive length mismatch. When length-normalised templates were being used for recognition, the input words were also normalised, and then no segmentation was required in the third stage of the recognition procedure. The use of length normalisation was found to make no significant difference to the recognition results.

Some preliminary experiments were conducted to determine appropriate values for the clustering parameters – namely, the number of clusters per word c, the small-initial-cluster exclusion size e, the presence or absence of postprocessing for single-token clusters, and the choice of DTW or linear averaging. In each case, the training data consisted of the first repetitions of the digits by each training speaker, and the test data consisted of all three repetitions by each of the test speakers. These experiments revealed that the value of e (in the range from 0 to 2), the use of postprocessing and the choice of the averaging method

made little difference to the recognition accuracy attained. With six templates per digit, the accuracy ranged from 92.8% to 93.1% (depending on the values of the other clustering parameters); with four templates per digit, it was 92.9%; with two templates per digit, derived by separate (DTW) averaging of the male and female training speakers' utterances, 91.7%; with one template per digit (constructed by averaging together the tokens from all 50 training speakers), again 91.7%. (However, in preliminary adaptive recognition experiments, it was found that the results with adaptation were better, by amounts ranging from 0.5% to 1.7%, for the set of two templates per digit than for a single template per digit.)

Three template sets were used in the main series of adaptation experiments: one (referred to as D6) consisting of six templates per digit, derived using $e = 2$, with postprocessing and DTW averaging; one (D4) containing four templates per digit ($e = 1$, postprocessing, DTW); and one (D2) containing two templates per digit, derived from separate averaging of the utterances of the male and female training speakers. In each case, the length normalisation was applied to the training data before the clustering procedure. For D4 only, length normalisation was also applied at an earlier stage, as preprocessing for the comparison of the training tokens to generate the table of distances for use in the clustering.

7.3: Adaptive recognition experiments and results

Several series of experiments with template adaptation were conducted using the data base described above. The recognition system and the experimental procedure were improved as the experiments progressed, to incorporate compensation for adaptation (by the mechanism described in section 5.3.5) and

random ordering of the input sequence for each test speaker. The experiments are described, and their results are presented and discussed, in sections 7.3.1-7.3.3 below.

Some of the speaker-independent template adaptation results without compensation (section 7.3.1) have appeared in a conference paper [258] which is reproduced at the end of this thesis. Results with compensation and random reordering (section 7.3.3), for one initial template set (D6), were published in a more recent paper [259], which is also attached. These and other results (for template set D2) are included in a further paper [260] which has been submitted for publication (as mentioned in section 6.3.2).

7.3.1: Experiments without compensation

In the first experiments conducted with adaptation of speaker-independent templates, no compensation factors were applied (as the provision for compensation factors had not yet been incorporated into *awr*). The input for each test speaker consisted of the three repetitions of the digits, in order, with the digits ordered from 0 to 9 within each repetition. This input sequence was recognised with and without adaptation; the differences between adaptive and non-adaptive recognition results on the three repetitions were computed, and averaged across the 49 test speakers. (This procedure is the same as the one-phase procedure described in section 6.3.1, and used in the first main series of speaker-specific template adaptation experiments in section 6.3.2.4. The two-phase procedure was not adopted for the speaker-independent template adaptation experiments because the small number (30) of utterances from each test speaker did not allow construction of adequate adaptation and evaluation data sets.) The results, for template sets D6 and D2, are given in tables 7.1 and 7.2. As in tables 6.5 to 6.8, the standard errors were estimated from the variations in the improvements

across test speakers. In these and all the other speaker-independent template adaptation experiments reported here, the template elimination thresholds used (t_1 and t_2) were 1.6 (after the first stage of comparison) and 1.12 (after the second stage). The accuracies without adaptation on the successive repetitions of the digits were, for D6, 92.45%, 94.49% and 92.04% respectively, and, for D2, 91.43%, 92.65% and 90.61%.

The notation for adaptation parameters in these tables is basically the same as in tables 6.5 to 6.10 (as described in section 6.3.2.4). An "s" after the adaptation parameters indicates "skewed" adaptation (as described in section 5.3); "+" indicates that the template for the second-best candidate word was adapted (positively or negatively as appropriate) when the best-matching template was incorrect.

The results with supervised adaptation in tables 7.1 and 7.2 show fairly significant improvements on the third repetitions of the digits by each test speaker, resulting from the prior adaptation to the first two repetitions. Without second-best-candidate adaptation, the improvement on the third repetition (averaged across the test speakers) ranged from 2.04% (D6, tracking, input weight 0.2) to 3.67% (D2, optimisation, initial input weight 1.0). With second-best adaptation, higher levels of improvement were attained, ranging from 4.69% to 6.74%; the significance of the improvements was also greater. (The confidences corresponding to the greatest third-repetition improvements without second-best adaptation are 0.998 (for D6) and 0.97 (D2); the confidences for the best cases with second-best adaptation are in excess of 0.9999 for both template sets.)

However, the overall improvement in accuracy, over all three repetitions of the vocabulary, was rather small, or in some cases negative, especially without second-best adaptation. This deficiency in the overall improvement with adapta-

Table 7.1: results of adaptive digit recognition experiments with speaker-independent initial templates (set D6)

Adaptation	Mean (standard error) of improvement over non-adaptive recognition				Mean (s.e.) overall recognition accuracy
	input repetition			overall	
	1	2	3		
none					92.99
					(1.00)
S t .2 -.05	-1.22	-0.20	2.04	0.20	93.20
	(0.47)	(0.74)	(0.77)	(0.36)	(1.07)
S t .2 -.05 +	-1.22	1.02	4.69	1.50	94.49
	(0.47)	(0.73)	(1.17)	(0.39)	(0.82)
S o .33 -.05	-2.04	-0.41	2.86	0.14	93.13
	(0.58)	(0.82)	(0.97)	(0.44)	(1.10)
S o .5 -.05	-2.24	-0.61	2.65	-0.07	92.92
	(0.60)	(0.98)	(1.12)	(0.63)	(1.22)
S o .67 -.05	-2.86	-0.82	2.86	-0.27	92.72
	(0.71)	(1.04)	(1.13)	(0.72)	(1.27)
S o 1.0 -.05	-2.04	-0.20	3.27	0.34	93.33
	(0.77)	(0.99)	(1.07)	(0.65)	(1.09)
S o .33 -.05 +	-2.45	1.22	5.31	1.36	94.35
	(0.69)	(0.69)	(1.24)	(0.51)	(0.90)
S o .5 -.05 +	-3.47	1.84	5.71	1.36	94.35
	(0.80)	(0.70)	(1.20)	(0.55)	(0.89)
S o .67 -.05 +	-3.88	1.84	5.51	1.16	94.15
	(0.87)	(0.70)	(1.20)	(0.57)	(0.98)
S o 1.0 -.05 +	-2.65	1.63	5.51	1.50	94.49
	(0.86)	(0.84)	(1.24)	(0.62)	(0.94)
U t .2 (1.15)	-1.02	-2.45	-1.02	-1.50	91.50
	(0.44)	(0.74)	(0.84)	(0.46)	(1.24)
U t .2 (1.1)	-1.22	-2.45	-1.63	-1.77	91.22
	(0.47)	(0.69)	(0.89)	(0.47)	(1.25)
U o .2 (1.15)	-1.02	-2.04	-0.61	-1.22	91.77
	(0.44)	(0.71)	(0.74)	(0.43)	(1.22)
U o .2 (1.15) s	-0.41	-1.43	0.00	-0.61	92.38
	(0.29)	(0.58)	(0.58)	(0.25)	(1.10)
U o .33 (1.15) s	-0.41	-2.86	-1.22	-1.50	91.50
	(0.29)	(0.82)	(0.86)	(0.38)	(1.21)

tion was due to the large decrease in accuracy which the adaptation caused on the first repetitions. Without second-best adaptation, this loss of accuracy on the first 10 digits ranged from 1.22% to 3.88%, and was highly significant across test speakers (with confidences exceeding 0.99, and in some cases exceeding 0.999);

Table 7.2: results of adaptive digit recognition experiments with speaker-independent initial templates (set D2)

Adaptation	Mean (standard error) of improvement over non-adaptive recognition				Mean (s.e.) overall recognition accuracy
	input repetition			overall	
	1	2	3		
none					91.56 (1.11)
S t .2 -.05	-1.22 (0.47)	0.82 (0.82)	2.86 (1.17)	0.82 (0.57)	92.38 (1.16)
S t .2 -.05 +	-1.43 (0.50)	3.06 (0.93)	5.31 (1.34)	2.31 (0.60)	93.88 (0.94)
S o .33 -.05	-2.45 (0.62)	-0.41 (1.01)	2.86 (1.30)	0.00 (0.79)	91.56 (1.40)
S o .5 -.05	-3.67 (0.86)	-1.63 (1.14)	2.45 (1.44)	-0.95 (0.86)	90.61 (1.49)
S o .67 -.05	-3.88 (0.87)	-0.41 (1.13)	2.86 (1.40)	-0.48 (0.82)	91.09 (1.40)
S o 1.0 -.05	-2.04 (0.77)	-1.22 (1.04)	3.67 (1.84)	0.95 (0.72)	92.52 (1.20)
S o .33 -.05 +	-2.65 (0.64)	2.86 (1.01)	6.12 (1.42)	2.11 (0.69)	93.67 (1.00)
S o .5 -.05 +	-4.90 (0.97)	2.86 (1.05)	6.74 (1.58)	1.56 (0.78)	93.13 (1.06)
S o .67 -.05 +	-5.51 (0.97)	3.06 (1.14)	6.33 (1.51)	1.29 (0.80)	92.86 (1.13)
S o 1.0 -.05 +	-3.67 (0.86)	2.86 (1.17)	6.74 (1.58)	1.97 (0.80)	93.54 (1.00)
U t .2 (1.15)	-0.41 (0.29)	-2.24 (0.98)	-3.06 (1.34)	-1.90 (0.70)	89.66 (1.41)
U t .2 (1.1)	-1.02 (0.44)	-2.24 (1.06)	-2.45 (1.32)	-1.90 (0.78)	89.66 (1.53)
U o .2 (1.15)	-0.41 (0.29)	-2.65 (0.86)	-2.24 (1.02)	-1.77 (0.56)	89.80 (1.34)
U o .2 (1.15) s	-0.61 (0.35)	-0.61 (0.45)	-1.63 (0.98)	-0.95 (0.39)	90.61 (1.18)
U o .33 (1.15) s	-1.22 (0.47)	-2.86 (0.97)	-3.06 (1.17)	-2.38 (0.62)	89.18 (1.40)

and there was also usually some loss in accuracy on the second 10-digit sequence. With second-best adaptation, the reduction in accuracy on the first repetition was from 1.22% to 5.51% (again with a high level of significance), but an improvement (ranging from 1.02% to 3.06%, and of moderate to high

significance) was attained on the second repetition. The poor performance on the first repetition of the digits can be explained by the lack of compensation for the unequal adaptation of the templates up to a given point in the input. As was explained in section 5.3, when an adapted template exists for one word but not for another, an utterance of the latter word is liable to be misrecognised as the former, because adapted templates tend to have smaller distances from input utterances than unadapted templates have. This is especially true when the initial (unadapted) templates are not specific to the current speaker: in this case, the correspondence of an adapted incorrect-word template to the speaker's voice characteristics may outweigh its lack of correspondence to the particular word spoken, resulting in a better match than is obtained for an unadapted correct-word template.

In the cases of supervised adaptation with the optimisation weighting, the best overall results tended to be attained with a small initial input weight (0.33, corresponding to twice as much weight on the initial template as on each input utterance) or a large one (1.0, corresponding to zero weight on the initial template) rather than with an intermediate weight value (0.5 or 0.67). The results with tracking, with input weight 0.2, were also relatively good. However, this preference for small or very large input weights could be a side-effect of the lack of compensation for the adaptation. It may be expected that the lack of compensation will degrade the performance more for weightings for which the optimal compensation factors are large.

The results with unsupervised adaptation were in all cases poorer than those with no adaptation – except that in one case, on the third repetitions of the digits, with skewed adaptation of template set D6, using the optimisation weighting with initial input weight 0.2, an accuracy equal to that without adaptation was achieved. The skewed form of adaptation reduced the loss of

accuracy relative to the direct form, but could not eliminate it completely. Again, the uneven adaptation effect, in the absence of compensation, may be blamed. With unsupervised adaptation, the system is less able to recover from errors on the first few input utterances than if the adaptation is supervised; adaptation to wrongly recognised inputs may occur, so that the accuracy on the affected words becomes progressively poorer, instead of better, as the adaptive recognition proceeds. The unsupervised adaptation results with adaptation threshold 1.1, and (especially) those with initial input weight 0.33, are worse than the corresponding results with threshold 1.15 or initial input weight 0.2: keeping the threshold high and the weight on the input small helps to restrict the effect of incorrect adaptations. (The difference between the results with thresholds 1.15 and 1.1 (with tracking in each case) is not significant: the confidences are 0.82 for D6 and 0.50 (corresponding to no difference between the two cases of threshold value) for D2. The difference between the results with weights 0.33 and 0.2 (with the optimisation weighting) is significant, however: the confidence associated with it is greater than 0.99 for each template set.)

7.3.2: Experiments with compensation and regular input order

In view of the observed phenomena with uneven adaptation, the compensation factor mechanism was incorporated into *awr*, and further experiments were conducted to determine the optimal compensation factors and the corresponding adaptive recognition results. The sets of compensation factors tested in these experiments are listed in table 7.3. (Cf. table 6.1.)

The first 13 sets of compensation factors in table 7.3 ("A" to "M") were designed for use in the supervised adaptation experiments. In these cases, compensation factors were defined only for templates adapted up to three times,

Table 7.3: compensation factors for speaker-independent template adaptation

Compensation code	Number of adaptations 1	2	3	4
A	1.3	1.45	1.55	
B	1.2	1.3	1.36	
C	1.15	1.22	1.27	
D	1.1	1.15	1.18	
E	1.17	1.25	1.3	
F	1.08	1.12	1.15	
G	1.2	1.25	1.28	
H	1.2	1.26	1.3	
I	1.1	1.14	1.16	
J	1.13	1.16	1.18	
K	1.15	1.19	1.22	
L	1.13	1.17	1.2	
M	1.11	1.155	1.18	
N	1.05	1.08	1.1	1.11
O	1.08	1.12	1.15	1.16
P	1.1	1.15	1.18	1.19
Q	1.15	1.19	1.22	1.23

since each test speaker's input included only three utterances of each digit, and no additional (incorrect) adaptations of a given template were permitted by the correctness condition imposed in the supervised adaptation. Sets "N" to "Q" were used in the unsupervised adaptation experiments, where incorrect adaptations could occur and so it was possible for a template to be adapted more than three times. (For templates adapted no more than three times, factors "O", "P" and "Q" are equivalent to factors "F", "D" and "K" respectively.)

The first speaker-independent template adaptation experiments with compensation were conducted using initial template set D6, with the utterances from each test speaker in the same order as before (i.e. three repetitions of the numerically increasing sequence from 0 to 9). Only two sets of (supervised) adaptation parameters were considered. The results are listed in table 7.4. For each of the two sets of adaptation parameters, the results are listed in order of

Table 7.4: results of adaptive digit recognition experiments with speaker-independent initial templates (set D6) and compensation factors

Adaptation and compensation		Mean (standard error) of improvement over non-adaptive recognition				Mean (s.e.) overall recognition accuracy
		input repetition			overall	
		1	2	3		
none						92.99 (1.00)
S o .5 0	none	-2.45 (0.62)	-0.82 (0.96)	2.65 (1.12)	-0.20 (0.64)	92.79 (1.25)
S o .5 0	F	-0.41 (0.50)	1.22 (0.81)	4.29 (0.92)	1.70 (0.46)	94.69 (0.89)
S o .5 0	D	0.00 (0.41)	0.82 (0.87)	4.29 (0.92)	1.70 (0.47)	94.69 (0.88)
S o .5 0	C	0.00 (0.41)	1.22 (0.81)	4.08 (0.87)	1.77 (0.42)	94.76 (0.85)
S o .5 0	E	0.20 (0.36)	1.02 (0.84)	4.08 (0.92)	1.77 (0.44)	94.76 (0.85)
S o .5 0	G	0.41 (0.29)	0.82 (0.82)	4.29 (1.05)	1.84 (0.51)	94.83 (0.79)
S o .5 0	B	0.41 (0.29)	0.82 (0.82)	3.88 (1.00)	1.70 (0.51)	94.69 (0.78)
S o .5 0	A	0.82 (0.40)	-0.20 (0.85)	0.41 (1.16)	0.34 (0.62)	93.33 (0.89)
S o 1.0 -.05	none	-2.04 (0.77)	-0.20 (0.99)	3.27 (1.07)	0.34 (0.65)	93.33 (1.09)
S o 1.0 -.05	F	-0.41 (0.58)	1.63 (0.89)	4.29 (1.05)	1.84 (0.52)	94.83 (0.85)
S o 1.0 -.05	I	-0.20 (0.54)	1.63 (0.98)	3.88 (1.08)	1.77 (0.61)	94.76 (0.80)
S o 1.0 -.05	D	-0.20 (0.54)	1.63 (0.98)	4.49 (1.09)	1.97 (0.60)	94.97 (0.77)
S o 1.0 -.05	J	0.41 (0.41)	1.84 (1.04)	3.47 (1.03)	1.90 (0.61)	94.90 (0.78)
S o 1.0 -.05	L	0.41 (0.41)	1.84 (1.04)	3.47 (1.03)	1.90 (0.61)	94.90 (0.78)
S o 1.0 -.05	K	0.61 (0.35)	1.22 (1.04)	2.86 (1.05)	1.56 (0.62)	94.56 (0.81)
S o 1.0 -.05	C	0.61 (0.35)	1.22 (1.04)	2.86 (1.09)	1.56 (0.62)	94.56 (0.80)
S o 1.0 -.05	G	0.61 (0.35)	0.61 (0.94)	1.63 (1.07)	0.95 (0.61)	93.95 (0.84)
S o 1.0 -.05	H	0.61 (0.35)	0.61 (0.94)	1.63 (1.07)	0.95 (0.61)	93.95 (0.84)

increasing compensation. (In the case with initial input weight 0.5, negative adaptation was (unintentionally) not performed, and so the results with no compensation are slightly different from those in table 7.1.)

It is apparent from these results that the use of appropriate compensation factors can improve adaptive recognition performance with speaker-independent initial templates, especially over the first two repetitions of the vocabulary in the input sequence. The optimal compensation factors are larger for the case with initial input weight 0.5 (where the performance without compensation was poorer) than for the case with initial input weight 1.0.

These results are not very realistic, however, in that the input for each test speaker consisted of three successive repetitions of the digits in order, rather than a randomly ordered digit sequence. With such regularly ordered input, it should be possible to obtain artificially high recognition accuracy by applying large compensation factors – since, whenever a word occurs in the input sequence for the nth time, each other word in the vocabulary has occurred either $n-1$ or n times already, and so the best templates for incorrect recognitions will mostly have been adapted, and hence be penalised by the compensation factors, at least as much as the best correct template. (This is most clearly true when there is only one template for each word of the vocabulary; the effect becomes blurred as the number of templates per word increases, since then the occurrence (and correct identification) of a given number of utterances of a word becomes less likely to result in that number of adaptations of the same template.) To obtain more realistic results, in further experiments, the order of each test speaker's input utterances was randomised before each recognition trial.

7.3.3: Experiments with randomly ordered input

For each of the 49 test speakers, four different random permutations of the 30-digit sequence were constructed. All the permutations for different test speakers were generated separately. In each trial, the randomly ordered sequence was recognised, with and without adaptation, using each of the three template sets (D6, D4 and D2). Differences between adaptive and non-adaptive performance were computed for the 10-digit input subsequences. For each test speaker, these differences were averaged across the four random orderings. Means and standard error estimates were obtained from the averaged differences for the 49 speakers.

The results of these experiments with randomly ordered input, for the three initial template sets, are shown in tables 7.5 to 7.7. Selected results (with no compensation, and with optimal compensation, for each set of adaptation parameters) are also plotted in figures 7.1 to 7.6.

7.3.3.1: Comparison of results with and without random ordering

Comparing the figures in tables 7.5 and 7.7 with the corresponding results for fixed-order input sequences in tables 7.1, 7.2 and 7.4, it is evident that, with adaptation but no compensation, the overall performance is usually slightly poorer for the randomly ordered input sequences. When the input consists of three successive repetitions of the vocabulary, the results with supervised adaptation show losses of accuracy on the first two repetitions, but a fairly significant improvement (averaging about 3%) on the third. (The contrast between the improvements on the first two repetitions and those on the third may be partly due to the difference in the non-adaptive recognition performance: the recognition accuracy without adaptation (given in section 7.3.1) was lower on the third

Table 7.5: results of adaptive digit recognition experiments with speaker-independent initial templates (set D6) and randomly ordered input sequences

Adaptation and compensation		Mean (standard error) of improvement over non-adaptive recognition				Mean (s.e.) overall recognition accuracy
		input subsequence			overall	
		1	2	3		
none						92.99 (1.00)
S o .5 0	none	-0.66 (0.46)	-1.22 (0.62)	-0.77 (0.80)	-0.88 (0.48)	92.11 (1.15)
S o .5 0	D	0.71 (0.38)	1.28 (0.43)	2.96 (0.64)	1.65 (0.56)	94.64 (0.81)
S o .5 0	K	0.92 (0.39)	1.28 (0.43)	2.91 (0.68)	1.70 (0.36)	94.69 (0.81)
S o .5 0	G	0.82 (0.30)	1.48 (0.49)	3.01 (0.73)	1.77 (0.24)	94.76 (0.79)
S o 1.0 -.05	none	0.46 (0.46)	-0.56 (0.64)	0.61 (0.77)	0.17 (0.47)	93.16 (0.95)
S o 1.0 -.05	D	1.12 (0.41)	1.12 (0.50)	2.70 (0.72)	1.65 (0.43)	94.64 (0.76)
S o 1.0 -.05	M	1.12 (0.41)	0.97 (0.51)	2.65 (0.73)	1.58 (0.44)	94.57 (0.77)
S o 1.0 -.05	K	0.20 (0.16)	0.15 (0.22)	0.26 (0.15)	0.20 (0.13)	93.20 (0.99)
S o 1.0 -.05	G	0.15 (0.17)	0.20 (0.23)	0.31 (0.21)	0.22 (0.14)	93.21 (0.99)
U o .2 (1.15)	none	-0.41 (0.21)	-1.53 (0.54)	-2.65 (0.73)	-1.53 (0.43)	91.46 (1.22)
U o .2 (1.15)	N	-0.10 (0.13)	-0.56 (0.37)	-0.46 (0.57)	-0.37 (0.28)	92.62 (1.10)
U o .2 (1.15)	O	0.10 (0.16)	0.05 (0.33)	0.10 (0.41)	0.09 (0.24)	93.08 (1.03)
U o .2 (1.15)	P	0.20 (0.19)	0.61 (0.35)	0.31 (0.45)	0.37 (0.26)	93.37 (0.97)
U o .2 (1.15)	Q	0.31 (0.24)	0.36 (0.41)	0.31 (0.56)	0.32 (0.32)	93.32 (0.91)
U o .2 (1.15) s	none	-0.26 (0.17)	-0.92 (0.30)	-0.92 (0.32)	-0.88 (0.19)	92.11 (1.05)
U o .2 (1.15) s	N	-0.05 (0.09)	0.31 (0.16)	0.41 (0.17)	0.22 (0.11)	93.21 (0.96)
U o .2 (1.15) s	O	0.20 (0.16)	0.31 (0.14)	0.26 (0.17)	0.26 (0.11)	93.25 (0.98)
U o .2 (1.15) s	P	0.20 (0.16)	0.15 (0.22)	0.26 (0.15)	0.20 (0.13)	93.20 (0.99)
U o .2 (1.15) s	Q	0.15 (0.24)	0.20 (0.41)	0.31 (0.56)	0.22 (0.14)	93.21 (0.99)

Table 7.6: results of adaptive digit recognition experiments with speaker-independent initial templates (set D4) and randomly ordered input sequences

Adaptation and compensation		Mean (standard error) of improvement over non-adaptive recognition: input subsequence 1	2	3	overall	Mean (s.e.) overall recognition accuracy
none						92.86 (1.02)
S o .5 0	none	-1.33 (0.46)	-1.84 (0.80)	-1.79 (0.84)	-1.65 (0.56)	91.21 (1.23)
S o .5 0	D	0.41 (0.37)	1.73 (0.64)	2.55 (0.78)	1.56 (0.45)	94.42 (0.81)
S o .5 0	K	0.82 (0.30)	2.09 (0.62)	2.65 (0.80)	1.85 (0.46)	94.71 (0.76)
S o .5 0	G	0.51 (0.23)	1.84 (0.71)	1.94 (0.77)	1.43 (0.47)	94.29 (0.78)
S o 1.0 -.05	none	-0.41 (0.39)	0.15 (0.71)	0.31 (0.91)	0.02 (0.53)	92.87 (0.97)
S o 1.0 -.05	D	0.26 (0.28)	1.99 (0.68)	2.40 (0.83)	1.55 (0.50)	94.40 (0.76)
S o 1.0 -.05	M	0.31 (0.27)	1.84 (0.74)	2.30 (0.83)	1.48 (0.51)	94.34 (0.76)
S o 1.0 -.05	K	0.31 (0.31)	1.63 (0.78)	2.14 (0.86)	1.36 (0.54)	94.22 (0.75)
S o 1.0 -.05	G	0.00 (0.33)	1.12 (0.84)	1.68 (0.83)	0.94 (0.57)	93.79 (0.71)
U o .2 (1.15)	none	-0.77 (0.41)	-1.53 (0.48)	-2.40 (0.69)	-1.56 (0.12)	91.29 (1.23)
U o .2 (1.15)	O	0.31 (0.19)	0.87 (0.31)	0.87 (0.45)	0.68 (0.25)	93.54 (0.92)
U o .2 (1.15)	P	0.46 (0.20)	0.77 (1.00)	0.97 (0.46)	0.80 (0.29)	93.66 (0.88)
U o .2 (1.15)	Q	0.15 (0.22)	1.02 (0.56)	0.36 (0.64)	0.51 (0.42)	93.37 (0.84)
U o .2 (1.15) s	none	-0.51 (0.23)	-1.48 (0.39)	-1.53 (0.54)	-1.17 (0.30)	91.68 (1.10)
U o .2 (1.15) s	O	-0.05 (0.05)	0.10 (0.22)	-0.10 (0.21)	-0.02 (0.10)	92.84 (1.02)
U o .2 (1.15) s	P	0.00 (0.00)	0.05 (0.19)	0.05 (0.17)	0.03 (0.08)	92.89 (1.02)
U o .2 (1.15) s	Q	0.00 (0.00)	-0.10 (0.19)	0.00 (0.22)	-0.03 (0.11)	92.82 (1.01)

Table 7.7: results of adaptive digit recognition experiments with speaker-independent initial templates (set D2) and randomly ordered input sequences

Adaptation and compensation		Mean (standard error) of improvement over non-adaptive recognition				Mean (s.e.) overall recognition accuracy
		input subsequence 1	2	3	overall	
none						91.56 (1.11)
S o .5 0	none	-0.71 (0.70)	-0.61 (0.94)	-1.43 (1.04)	-0.92 (0.74)	90.64 (1.35)
S o .5 0	D	1.07 (0.46)	3.21 (0.74)	3.01 (0.61)	2.43 (0.49)	94.00 (0.99)
S o .5 0	K	1.22 (0.43)	2.91 (0.69)	3.16 (0.69)	2.43 (0.48)	94.00 (0.90)
S o .5 0	G	1.12 (0.41)	2.76 (0.79)	3.11 (0.75)	2.33 (0.51)	93.90 (0.86)
S o 1.0 -.05	none	0.20 (0.56)	0.87 (0.86)	0.41 (0.86)	0.49 (0.65)	92.06 (1.11)
S o 1.0 -.05	D	1.12 (0.43)	2.81 (0.79)	3.52 (0.74)	2.48 (0.54)	94.05 (0.86)
S o 1.0 -.05	M	1.22 (0.45)	2.76 (0.79)	3.57 (0.76)	2.52 (0.54)	94.08 (0.84)
S o 1.0 -.05	K	0.87 (0.30)	2.09 (0.65)	2.30 (0.54)	1.75 (0.38)	93.32 (0.93)
S o 1.0 -.05	G	0.97 (0.32)	1.84 (0.61)	2.04 (0.55)	1.62 (0.37)	93.18 (0.93)
U o .2 (1.15)	none	-0.66 (0.33)	-1.79 (0.63)	-3.88 (0.93)	-2.11 (0.52)	89.45 (1.37)
U o .2 (1.15)	N	0.31 (0.25)	1.58 (0.40)	0.82 (0.50)	0.90 (0.29)	92.46 (1.09)
U o .2 (1.15)	O	0.56 (0.29)	1.33 (0.37)	0.97 (0.48)	0.95 (0.29)	92.52 (1.03)
U o .2 (1.15)	P	0.77 (0.34)	1.58 (0.42)	1.12 (0.46)	1.16 (0.31)	92.72 (1.01)
U o .2 (1.15)	Q	0.77 (0.35)	1.99 (0.51)	1.53 (0.51)	1.43 (0.34)	92.99 (0.93)
U o .2 (1.15) s	none	-0.20 (0.22)	-0.51 (0.47)	-2.24 (0.62)	-0.99 (0.35)	90.58 (1.16)
U o .2 (1.15) s	N	0.10 (0.10)	0.82 (0.30)	0.41 (0.29)	0.44 (0.16)	92.01 (1.13)
U o .2 (1.15) s	O	0.05 (0.14)	1.48 (0.41)	0.71 (0.26)	0.75 (0.22)	92.31 (1.11)
U o .2 (1.15) s	P	0.20 (0.18)	1.58 (0.44)	0.97 (0.29)	0.92 (0.25)	92.48 (1.09)
U o .2 (1.15) s	Q	0.20 (0.16)	1.28 (0.38)	0.77 (0.27)	0.75 (0.21)	92.31 (1.12)

repetition than on the first or second, for each of the template sets D6 and D2, leaving more room for improvement through adaptation.) With randomly ordered input, however, the results in the case with initial input weight 0.5 show losses of accuracy spread across all three 10-word input subsequences; and, while improvements were attained in the case with initial input weight 1.0, these were relatively small (for all subsequence numbers) and of low significance. This confirms, for the case with no compensation, the expectation that the benefit of adaptation should be greatest if the adaptation proceeds evenly across the words of the vocabulary. (The differences in overall accuracy between the two cases of input ordering are not highly significant; but there are significant differences in the improvements on the third 10-digit subsequences (confidences 0.996 and 0.998 for D6 with the two sets of supervised adaptation parameters, and 0.997 and 0.994 for D2) – though part of the difference here may be due to the effect (already mentioned) of the non-adaptive recognition accuracy on the improvement attainable through adaptation in the case without random reordering.)

In the case of the template set (D6) with which the previous compensation experiments were conducted, the overall improvements due to adaptation with appropriate compensation are somewhat smaller with the randomly ordered input – 1.77% and 1.65% for the two supervised adaptation parameter settings (table 7.5), against 1.84% and 1.97% respectively (table 7.4). However, the significance of this difference is very low. Even on the third subsequences, where the difference in the degree of improvement is greatest, it is only weakly significant (with means 1.28 and 1.79, and standard errors 1.06 and 1.15, giving confidences 0.88 and 0.94, for the respective adaptation parameters). It appears, then, that there is less difference between results with regular and random ordering with appropriate compensation than in the absence of compensation.

Figure 7.1: results for template set D6 with supervised adaptation

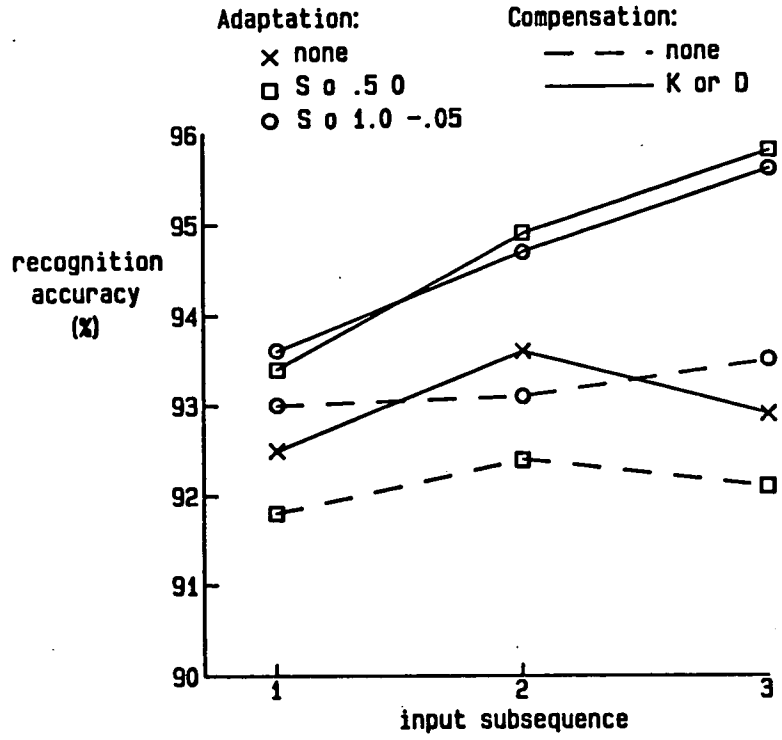


Figure 7.2: results for template set D6 with unsupervised adaptation

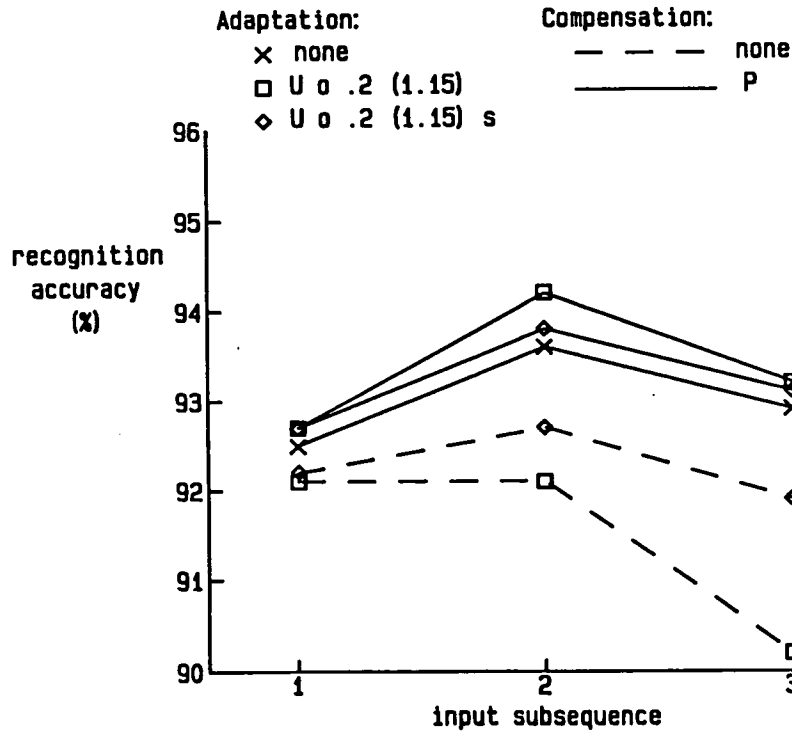


Figure 7.3: results for template set D4 with supervised adaptation

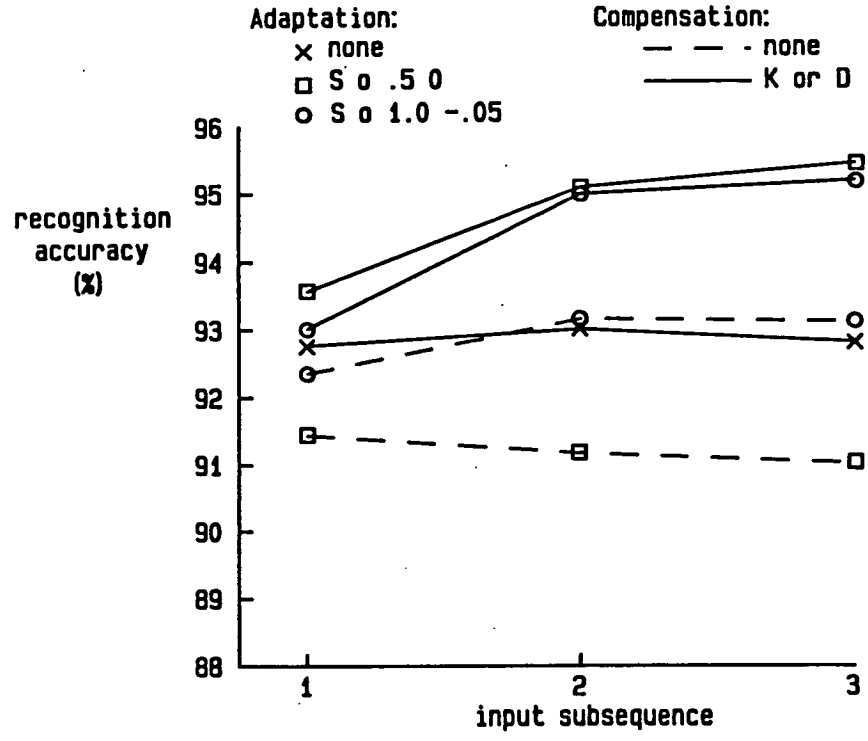


Figure 7.4: results for template set D4 with unsupervised adaptation

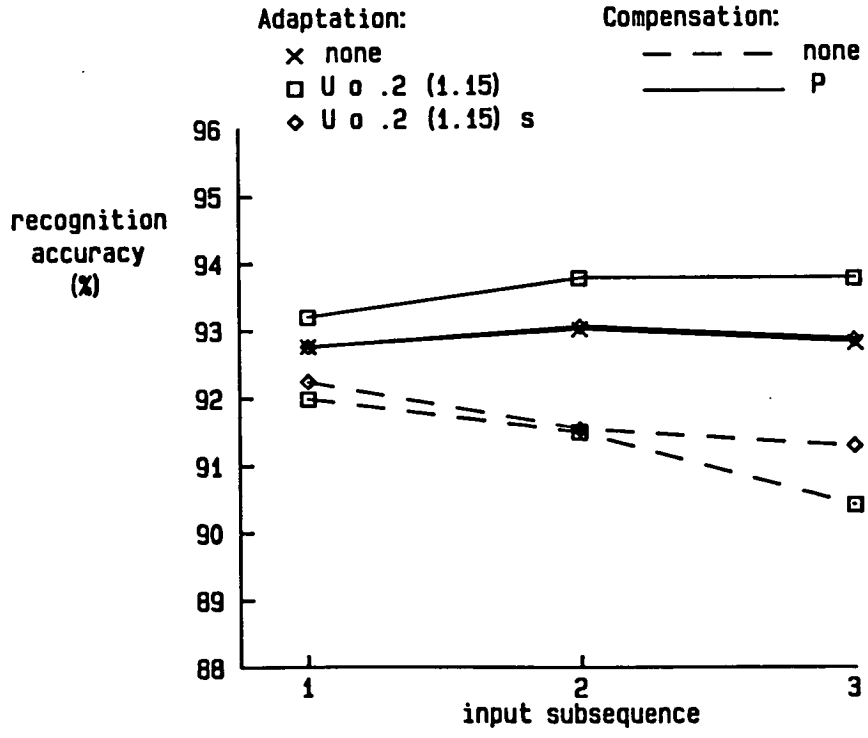


Figure 7.5: results for template set D2 with supervised adaptation

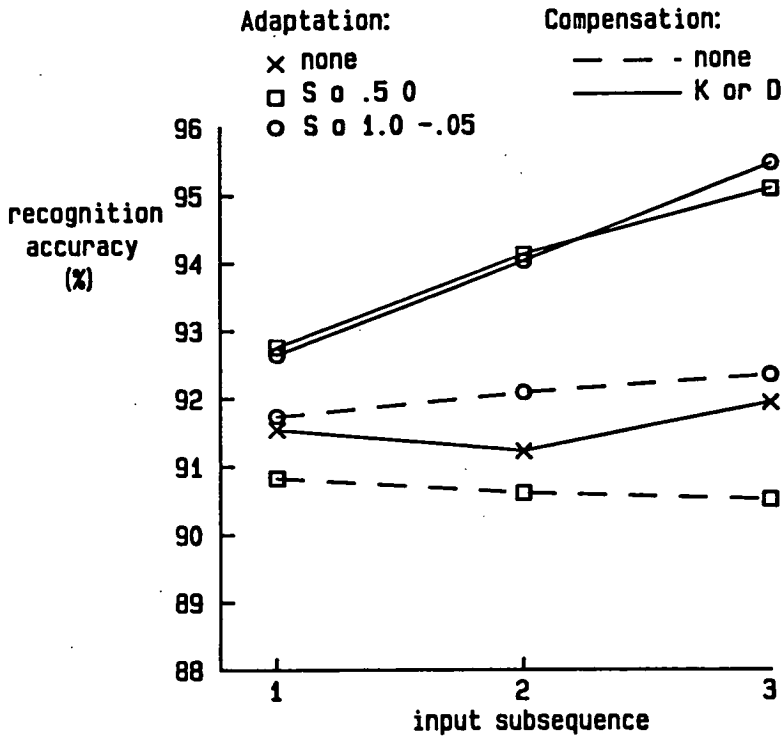
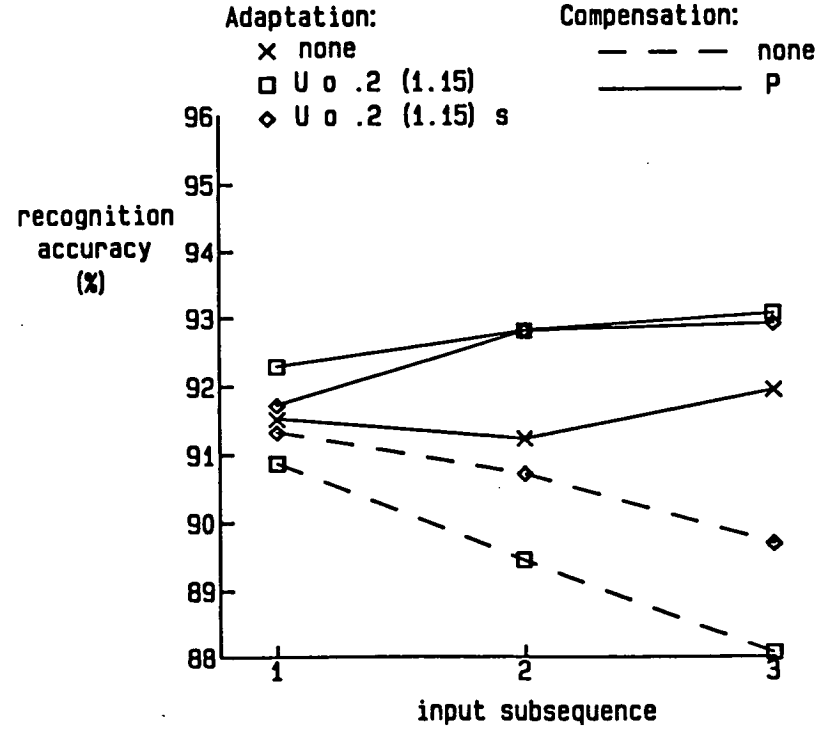


Figure 7.6: results for template set D2 with unsupervised adaptation



7.3.3.2: Observations on the supervised adaptation results

With appropriate compensation, supervised adaptation yields substantial improvements in recognition accuracy, with high levels of statistical significance across the test speakers. The accuracies on the third 10-digit subsequences were improved, with the optimal adaptation and compensation, by amounts greater than 2.5%, and accuracies between 95% and 96% were attained on these subsequences, for all three initial template sets. The confidences for the third-subsequence improvements, with input weight 0.5 and compensation "K", are better than 0.9999 for template sets D6 and D2, and 0.9992 for D4. (The improvements with input weight 1.0 and compensation "D" were of fairly similar size and significance.) Fairly significant, though smaller, improvements were attained even on the first 10-digit subsequences of the input: the confidences in these improvements, again with input weight 0.5 and compensation "K", are 0.99 (D6), 0.995 (D4) and 0.997 (D2).

As in the experiments without reordering of the input, it is noticeable in tables 7.5 to 7.7 that the optimal compensation factors are larger for the case with initial input weight 0.5 (equally weighted averaging of the initial template and input utterances) than for the case with initial input weight 1.0 (zero weight on the initial template). For each template set, with initial input weight 0.5, the overall improvement with compensation "K" is greater than that with compensation "D", whereas this inequality is reversed in the case with initial input weight 1.0. Using results averaged (for each test speaker) across the three initial template sets, the preference for compensation "D" over compensation "K" with initial input weight 1.0 is at least fairly significant on each of the three input subsequences (and highly significant on the third), and has a high level of significance overall (mean 0.79, standard error 0.18, confidence greater than

0.9999); the preference for compensation "K" over compensation "D" with initial input weight 0.5 is less significant (mean 0.11, standard error 0.11, confidence 0.84); and the difference between the preferences for the two cases of weighting is also highly significant overall (mean 0.90, standard error 0.22, confidence 0.9999) and is moderately to highly significant on the individual subsequences.

The fact that adaptation with initial input weight 0.5 has larger optimal compensation factors than adaptation with initial input weight 1.0 indicates that the reduction of word distances due to adaptation is more rapid in the former case. That is, over short input sequences such as those used in these experiments, averaging the speaker-independent initial template with the inputs recognised so far results in templates more closely matching the current speaker's speech than replacing the initial template completely by the average of the speaker-specific recognised input utterances (which after one adaptation is just a single utterance). (The effect is seen most clearly in the case of template set D6, which is the best of the three initial template sets: with this template set, the overall accuracy with initial input weight 1.0 drops sharply, from 94.57% to 93.20%, when the compensation factors are increased from the "M" values to the "K" values, whereas compensation "K" yields good results with initial input weight 0.5.) This phenomenon was not, however, accompanied by a significant difference between the recognition accuracies attainable (with suitable compensation factors) with the two weight settings. The mean difference (in overall accuracies averaged across the three template sets) between results with initial input weight 1.0 and compensation "D" and those with initial input weight 0.5 and compensation "K" is 0.10, and the standard error is 0.12, yielding a confidence of only 0.79 that the latter case is better. The corresponding results for the individual template sets are likewise inconclusive. (It should be remembered, however, that negative adaptation was in operation in the case

with initial input weight 1.0, and not in the case with initial input weight 0.5. If the negative adaptation had been consistent across the two cases, the difference in accuracy between them might have been slightly larger and more significant.)

The amount of improvement attained over non-adaptive recognition was greatest for initial template set D2 (where the maximal overall improvement was 2.52, from 91.56% to 94.08%), next greatest for D4 (1.85, from 92.86% to 94.71%), and least for D6 (1.77, from 92.99% to 94.76%). This is as expected: the fewer templates per word in a speaker-independent template set, the less well it covers the range of variant pronunciations and voices; hence, the lower its accuracy will tend to be (without adaptation), and the more room there will be for improvement through adaptation. The differences between corresponding results with different template sets are not very consistent across test speakers, however. The differences between non-adaptive results for D6 and D2 and for D4 and D2 are fairly significant (respectively, 1.43 (standard error 0.67, confidence 0.98) and 1.29 (0.62, 0.98)), but the difference of D6 and D4 is not significant. With adaptation (and appropriate compensation), the differences between different template sets' results are only moderately significant at best (the mean does not exceed 1.5 standard errors, and so the confidence does not exceed 0.93): the distributions of accuracy (across test speakers) for the different template sets have moved closer together and so overlap considerably. However, the change, due to adaptation, in the difference between template sets' accuracies is also not highly significant: its mean value is, for each of the template set pairs (D6,D2) and (D4,D2), between 1.0 and 1.5 times its standard error (and for (D6,D4) it is less than one standard error). (This "change in the difference" is also, by a trivial rearrangement of subtractions, the difference between the accuracy improvements due to adaptation for the different template sets.) No

improvement in significance is obtained by considering results on the third subsequences instead of overall results.

The effects of supervised adaptation on the accuracies for individual test speakers are shown by the histograms in figures 7.7 and 7.8. In each of these figures, the histograms in the first column show results for template set D6; those in the second column, for D4; those in the third column, for D2. The quantities whose frequency distributions (over the 49 test speakers) are shown in the first two histograms in each column are the numbers of errors in the non-adaptive and adaptive recognition of the third 10-digit subsequences, accumulated over the four random orderings of the input data for each test speaker. These numbers are counts of errors occurring in 40 recognitions, and can therefore be converted to percentage error rates for individual test speakers by multiplication by 2.5. The third histogram in each column shows the distribution of the improvement in accuracy due to adaptation, again accumulated over the final 10-digit subsequences in the four random orders for each speaker. Thus the quantity appearing in the third histogram, for each test speaker, is the difference between the error counts for that speaker which contribute to the two histograms above it. These error counts and differences are shown in figure 7.7 for adaptation with initial input weight 0.5 and compensation "K" and in figure 7.8 for adaptation with initial input weight 1.0 and compensation "D". (For each template set, the histograms of error counts for non-adaptive recognition in figures 7.7 and 7.8 – and indeed in figures 7.9 and 7.10 – are identical: they are repeated in the different figures for ease of comparison with the histograms for adaptive recognition.)

The error count histograms illustrate both the improvements in mean recognition accuracy due to supervised adaptation with appropriate compensation (seen in the "subsequence 3" columns of tables 7.5 to 7.7) and the reductions

Figure 7.7: histograms of individual test speakers' results
with supervised adaptation (parameters S o .5 0; compensation "K")

Results in each column:
error counts without adaptation;
error counts with adaptation;
reductions in error count due to adaptation
- computed over third 10-digit subsequences in all random orders

D6 D4 D2

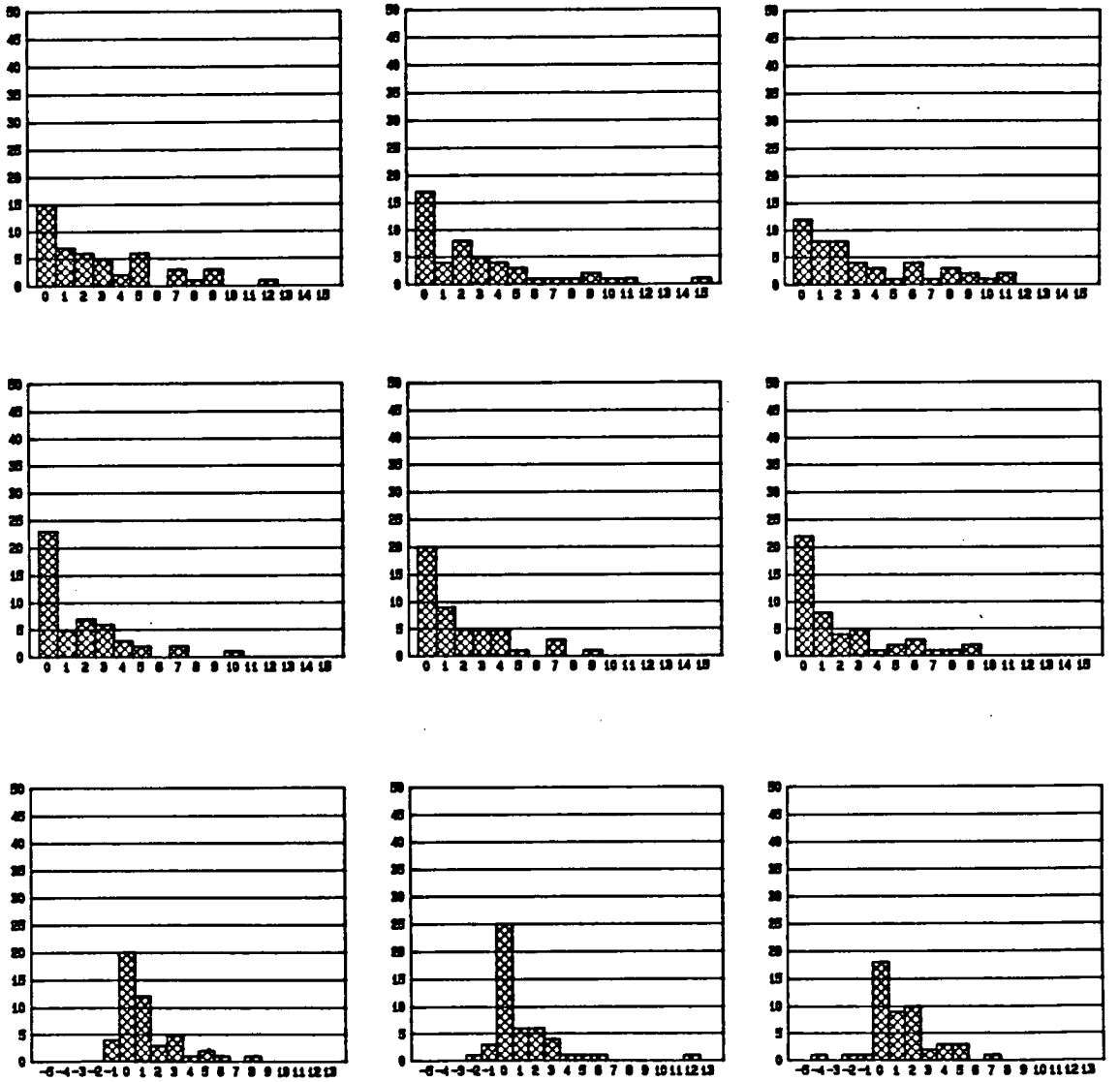
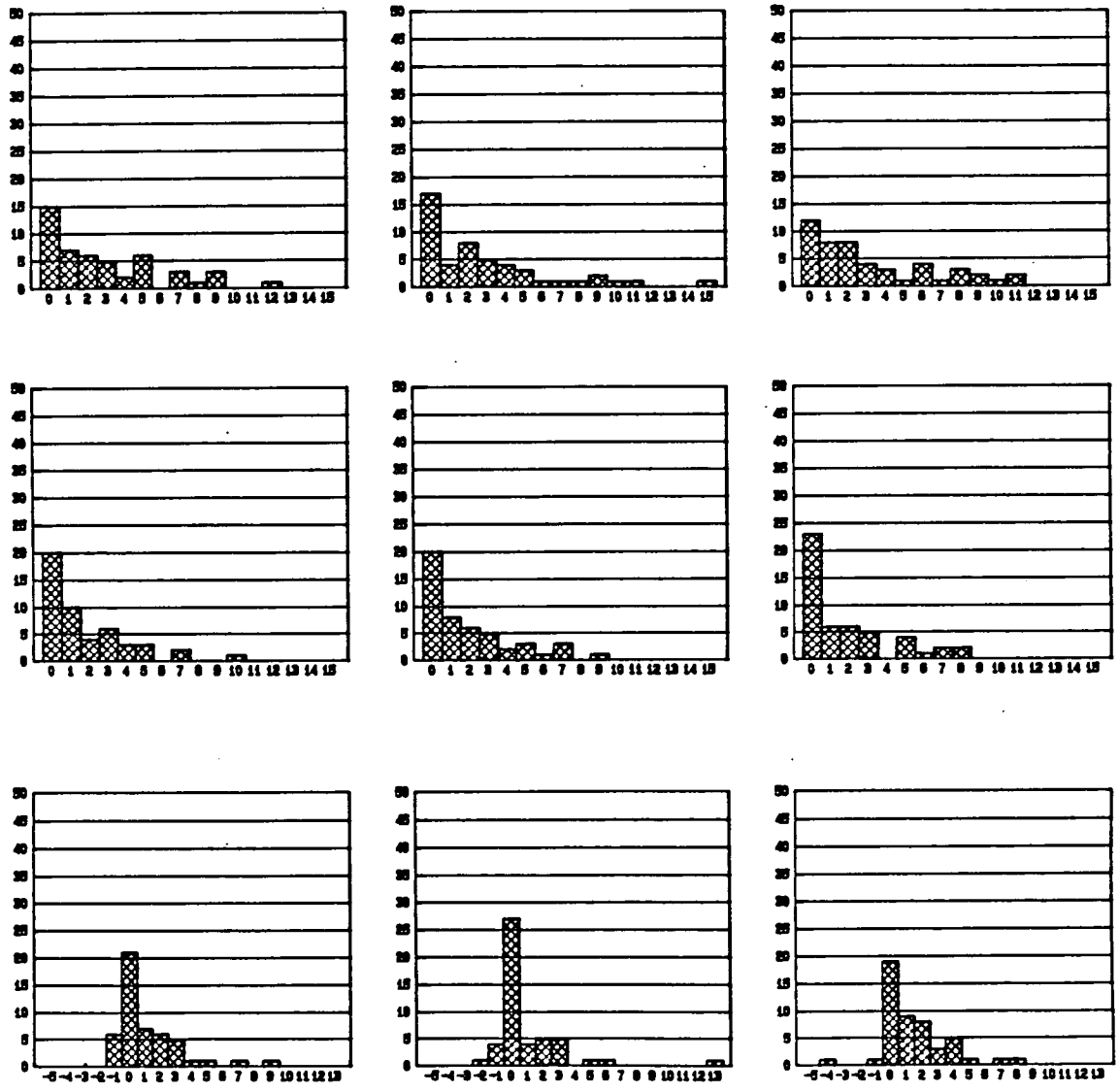


Figure 7.8: histograms of individual test speakers' results
with supervised adaptation (parameters S o 1.0 -.05; compensation "D")

Results in each column:
error counts without adaptation;
error counts with adaptation;
reductions in error count due to adaptation
- computed over third 10-digit subsequences in all random orders

D6 D4 D2



in the variation of accuracy across speakers (which are revealed, for the full 30-digit sequences, in the reduced standard error estimates for the overall accuracies in the final columns of tables 7.5 to 7.7). For each template set, the tail of high error rates for particular speakers in the histogram for the non-adaptive case is shortened, and the occurrence of zero error counts is increased, in the histogram for each adaptive case. The difference histograms show that, for each template set and adaptation weighting, there were only a few of the 49 speakers (never more than six of them) whose third-subsequence results in adaptive recognition were poorer than the results on the same data in non-adaptive recognition, and the loss of accuracy in such cases was usually small (amounting to one recognition out of 40, or occasionally two) – whereas a large proportion of the test speakers (often more than half of them) had improvements in accuracy, which were sometimes very substantial (e.g. five more correct recognitions out of 40, an improvement of 12.5%).

7.3.3.3: Observations on the unsupervised adaptation results

Without compensation, the results with unsupervised adaptation, like those in tables 7.1 and 7.2, show only losses of accuracy relative to the performance without adaptation. Moreover, the loss in accuracy generally increases from one input subsequence to the next. With direct adaptation (i.e. not skewed), the reduction in overall accuracy, and in accuracy on the third subsequence, is greater for template set D2 than for D4 and D6 – though this difference has only moderate significance (confidences 0.84 and 0.87 for the respective differences in the overall results, and 0.96 and 0.93 for those on the third subsequences). With skewed adaptation, the reduction in accuracy is smaller than with the direct form, especially for D2, but is still highly significant. The

confidences for the reductions in overall accuracy, derived from results averaged across the three template sets, are greater than 0.9999 for both direct and skewed adaptation, and the same is true for the reductions in accuracy on the third subsequences.

With compensation, however, some improvement over non-adaptive recognition is attained. The improvement is greater for direct adaptation than for skewed adaptation – especially for template sets D4 and D2. Using results averaged across the three template sets, the mean improvements with compensation "P" are 0.78 (standard error 0.18, confidence > 0.9999) with direct adaptation and 0.39 (0.09, > 0.9999) with skewed adaptation; the difference between the direct and skewed adaptation results is 0.39, with standard error 0.18, and hence confidence 0.98. Thus, the improvement over non-adaptive recognition is highly significant overall for each form of unsupervised adaptation, and the difference between the improvements with direct and skewed adaptation is fairly significant. The improvements, averaged across the three template sets, with compensation "Q" are slightly smaller than those with compensation "P". The confidences in the overall improvements with direct adaptation and compensation "P" for the individual template sets are 0.92 (for D6), 0.995 (for D4) and 0.9998 (for D2); the confidences in the improvements with skewed adaptation are 0.93, 0.65 and 0.9997 respectively.

The rate of incorrect adaptations (relative to all recognitions, over all three template sets) with no compensation is 2.6% for direct adaptation, and 1.8% for skewed adaptation; with compensation "P", these rates are reduced to 1.3% and 1.1% respectively.

It is difficult to tell from the results of these experiments, where the length of each input sequence was limited to 30 digits (three repetitions of the vocabulary), what the outcome of unsupervised adaptation would be for long input

sequences. It seems likely that instabilities would set in, through repeated adaptation to misrecognised inputs, in some cases. This would probably tend to happen more with direct adaptation than with skewed adaptation (in which the effect of an incorrect adaptation is moderated because the adaptation is not applied to the best-matching template), and so, over long sequences of input utterances, the performance with skewed adaptation might prove to be better than that with direct adaptation. However, for many applications of speech recognition, the user will not tolerate repeated failure to recognise a particular word, and will prefer to retrain the templates for the affected words whenever such an instability occurs. The assessment of system performance is less straightforward if a retraining facility is assumed. It might be better to use direct adaptation, despite the risk of instabilities, if these could be corrected easily and the recognition performance was otherwise better than with skewed adaptation.

In the case of template set D2 particularly, it was to be expected that the improvement with skewed adaptation would be less than that with direct adaptation, since this template set contains one male and one female template for each word. When, for instance, a word spoken by a female test speaker is recognised correctly – the best-matching template being the one formed from the female training speakers' utterances – the adaptation is applied to the template formed from the male training speakers' utterances. It may take several such adaptations before the adapted (originally male) template matches the test speaker's voice well enough to contribute usefully to the recognition.

For a complete study of unsupervised adaptation, it would be desirable to obtain results with several different adaptation weightings, and with several values of the distance ratio threshold used in the adaptation condition. It might be found that a smaller input weight, or a higher threshold on the distance

ratio, would prevent many of the incorrect adaptations, while retaining an adequate rate of correct adaptation, and so yield an improvement greater than was obtained with the weighting and threshold value adopted here. However, these experiments have demonstrated that, with appropriate compensation factors, it is possible to improve the overall recognition accuracy of an initially speaker-independent system by unsupervised adaptation.

Histograms of the recognition errors on the third 10-digit subsequences, and of the improvements (on the third subsequences) due to the unsupervised adaptation, for individual test speakers are presented in figures 7.9 and 7.10. The layout of these figures is the same as that of figures 7.7 and 7.8. Figure 7.9 shows the error counts and improvements with direct adaptation, and 7.10 those with skewed adaptation.

One noticeable feature of the difference histograms in figures 7.9 and 7.10, when they are compared with those in figures 7.7 and 7.8, is that, when the adaptation is unsupervised, considerably larger numbers of test speakers have their recognition accuracies unchanged by it. This is especially true in the case of skewed adaptation applied to template sets D6 and D4. With direct unsupervised adaptation, the incidence of worsenings in performance for individual test speakers is only a little greater than with supervised adaptation, but there are substantially fewer cases of improvement, and the amount of improvement when it does occur tends to be smaller. Given the shortness of the test input sequences, this lesser degree of improvement may be partly because of the smaller input weight which was adopted (to reduce the danger of instability) in the unsupervised adaptation, as well as because of the reduced number of correct adaptations (resulting from the distance ratio threshold condition) and the counterproductive effect of incorrect adaptations. With skewed adaptation, the amount of improvement observed for any one speaker is further reduced (never

Figure 7.9: histograms of individual test speakers' results
with unsupervised adaptation (parameters U o .2 (1.15); compensation "P")

Results in each column:
error counts without adaptation;
error counts with adaptation;
reductions in error count due to adaptation
- computed over third 10-digit subsequences in all random orders

D6 D4 D2

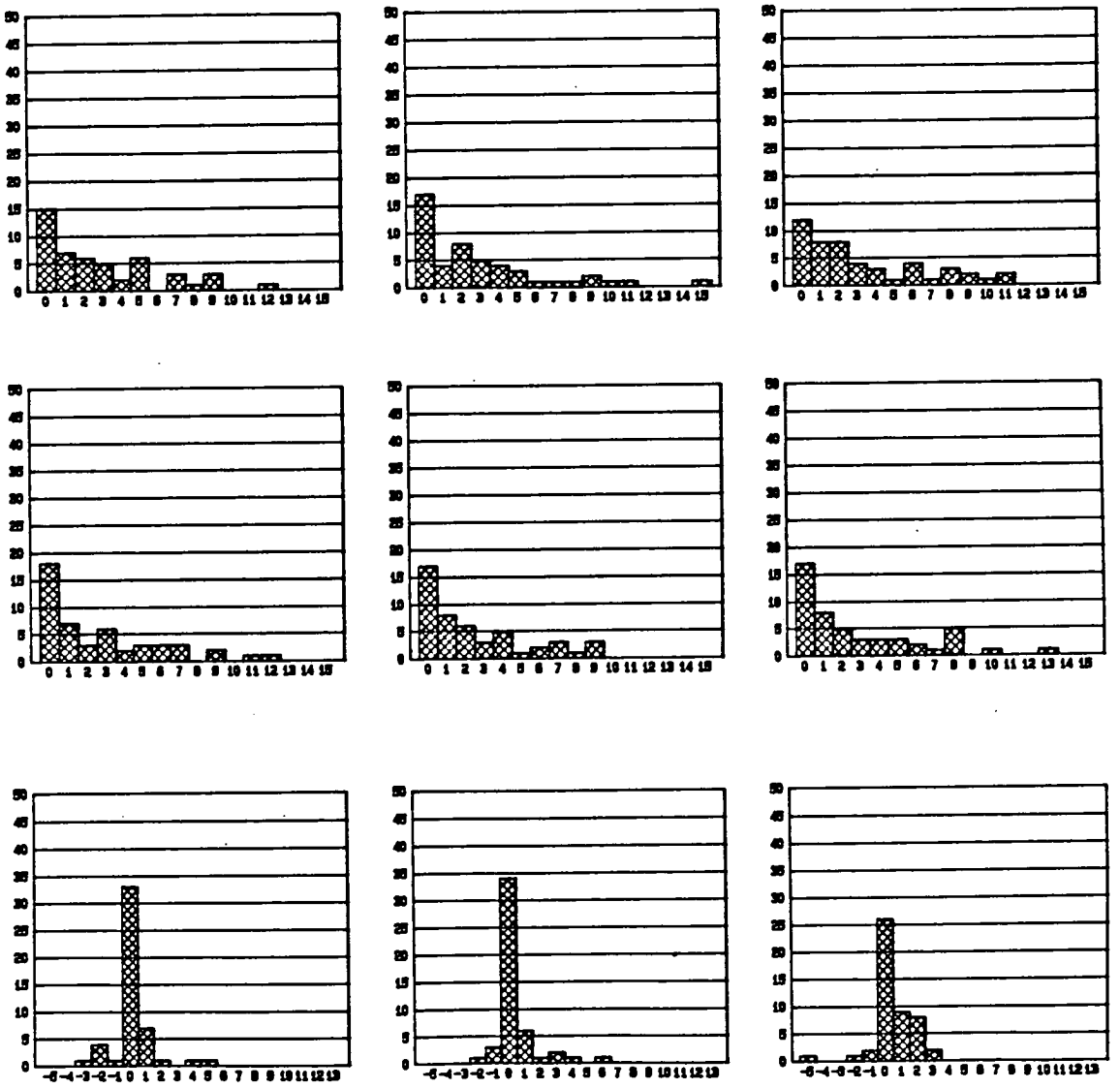
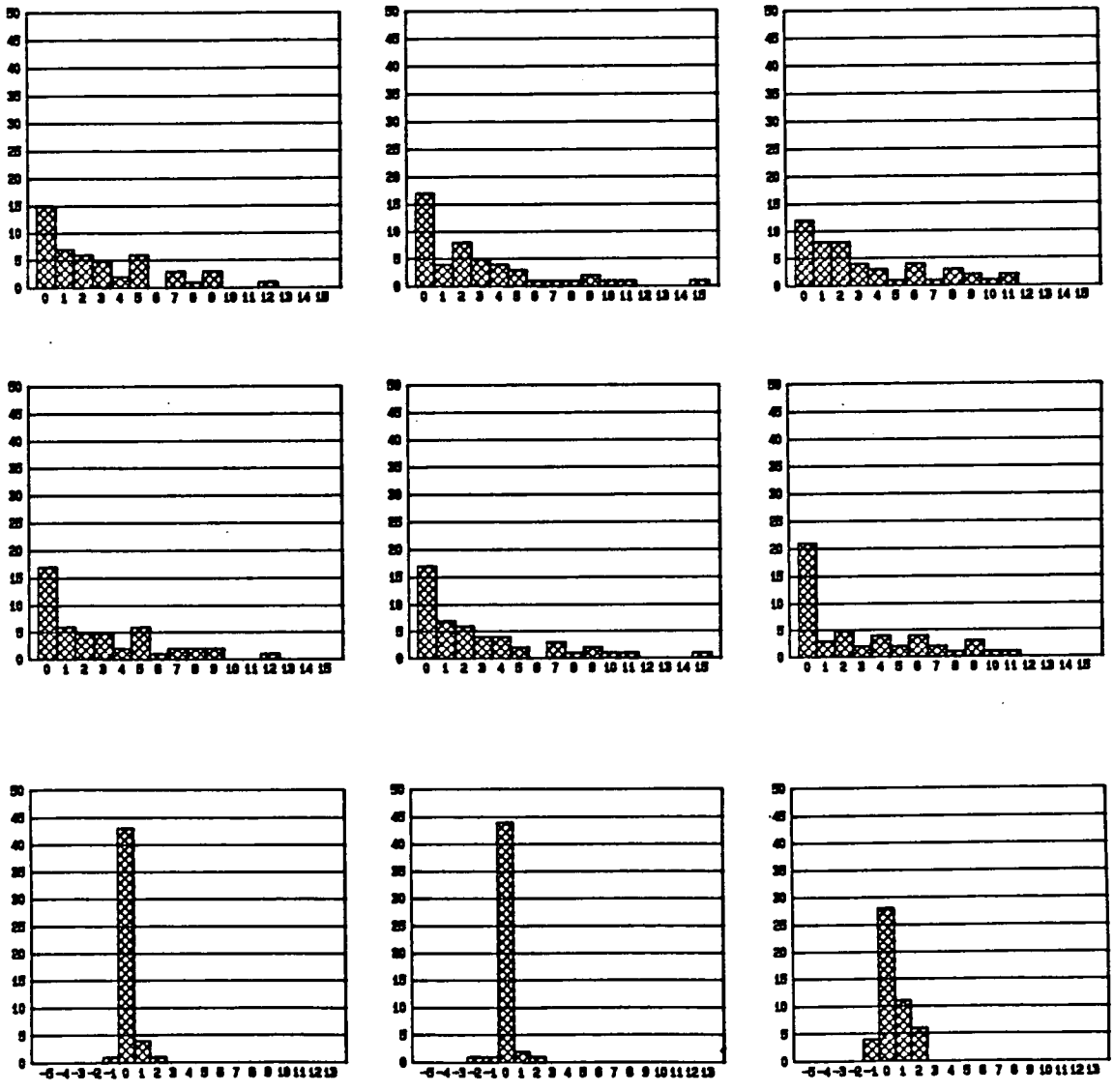


Figure 7.10: histograms of individual test speakers' results
with unsupervised adaptation (parameters U o .2 (1.15) s; compensation "P")

Results in each column:
error counts without adaptation;
error counts with adaptation;
reductions in error count due to adaptation
- computed over third 10-digit subsequences in all random orders

D6 D4 D2



exceeding a difference of two recognitions out of the 40), but the amount of deterioration that occurs is also restricted (usually to a loss of one recognition out of 40).

Overall, with unsupervised adaptation, the shortening of the tail of poor single-speaker results (seen in the non-adaptive error histograms) is much less evident than with supervised adaptation. Indeed, in one case (D2, direct adaptation), the worst error rate in the population was increased by the adaptation (from 11 to 13 out of 40 – though the speaker with 13 errors was one who had only eight errors without adaptation); and in only one case (D4, direct adaptation) was it reduced (from 15 to nine out of 40). A problem with unsupervised adaptation is that if a speaker's pronunciations of certain words do not correspond well enough to the templates to allow recognition with a word distance ratio above the threshold then these words' templates will never be adapted and the error rate for that speaker will remain high.

7.3.3.4: The effect of adaptation on computational requirements

As was mentioned in section 6.3.2.5, template adaptation can reduce the numbers of templates matched against the input at the second and third stages of the three-stage recognition procedure, and so improve the computational efficiency of the recognition. This effect is greater in the case of an initially speaker-independent template set, since the numbers of templates retained at the second and third stages in the non-adaptive case are larger than with speaker-specific templates – especially when there are several templates per word of the vocabulary.

Table 7.8: Numbers of template matches at the second and third stages (per recognition) in digit recognition with speaker-independent initial templates (during third subsequences of input)

Adaptation and compensation		Template set: D6 stage 2	D6 stage 3	D4 stage 2	D4 stage 3	D2 stage 2	D2 stage 3
none		12.52	1.02	9.41	0.88	6.52	0.66
S o .5 0	K	5.68	0.38	3.92	0.30	2.39	0.23
S o 1.0 -.05	D	4.63	0.40	3.23	0.29	2.07	0.26
U o .2 (1.15)	P	10.12	0.78	7.37	0.65	4.65	0.47
U o .2 (1.15) s	P	11.68	0.94	8.65	0.76	5.34	0.54

The average numbers of templates matched at the second and third stages during recognition of the third 10-digit subsequence of each test speaker's input, in cases without adaptation and with adaptation (and optimal compensation), are listed in table 7.8. These numbers were taken from the output of the experiments with randomly-ordered 30-digit input sequences. Thus, the numbers for the cases with adaptation show the effects of the preceding adaptation to the first 20 input utterances, and also of the continuing adaptation during the third subsequence.

With normalisation of all utterances to 30 interpolated vectors (which was applied in these experiments), the computation for one template adaptation is somewhat greater than that for a template match at stage 3, since the adaptation process involves a DTW alignment and some weighted averaging and interpolation operations. The DTW alignment is the most computationally costly part of the adaptation, however, and so the computation for an adaptation will not be as much as twice that for a stage 3 matching operation. A stage 3 match requires about eight or nine times as much computation as a (10-vector) template match at stage 2. Thus, in the case with the greatest recognition computa-

tion reduction (relative to non-adaptive recognition) in table 7.8 (D6, supervised adaptation, initial input weight 1.0), the computational saving during the recognition (7.89 matches per recognition at stage 2 and 0.62 at stage 3) is probably sufficient to outweigh the additional computation required for the adaptation. The same may be true for the other cases with initial templates D6 or D4 and supervised adaptation. With the initial template set D2, or with unsupervised adaptation, however, the computation reductions observed are outweighed by the computation for the adaptation. Greater reductions in computation during the recognition process can be expected after more prolonged adaptation, as the templates become more perfectly attuned to the voice and pronunciations of the speaker, and so it may be expected that, with a speaker-independent set of four or more templates per digit (such as D4 or D6), supervised adaptation will reduce the overall computation over long sequences of input utterances.

In general, the number of templates per word at which the long-term computational benefit of adaptation begins to outweigh its computational cost will depend on the confusability of the vocabulary: for a more confusable vocabulary, in which more templates tend to be retained at the second and third stages, the adaptation will pay off with fewer templates per word than for a less confusable vocabulary.

It may be noted in passing that the figures for non-adaptive recognition in table 7.8 show a beneficial effect of multiple-stage recognition in a system with several templates per word: although the total number of templates in use is twice as large in the case of D4 as in the case of D2, and three times as large in the case of D6, the numbers of templates retained at the second and third stages do not increase proportionately.

7.4: Discussion of speaker-independent template adaptation

Various features of the results have been examined in detail in the preceding sections (7.3.3.1 to 7.3.3.4). A few of the more general findings are discussed briefly in this section, and some issues relating to the evaluation and application of speaker-independent template adaptation are considered.

The main conclusions of these experiments with adaptation of speaker-independent templates are that the recognition performance can be substantially and significantly improved by supervised adaptation and that it can also be improved in general, though not so rapidly and not so consistently across initial template sets and test speakers, by unsupervised adaptation. By adaptation, templates which initially correspond rather poorly to the current speaker's pronunciations of the words, because they have been formed from utterances by a standard set of training speakers (not including this speaker), can be made speaker-specific during the course of a recognition session, and the recognition accuracy can be correspondingly increased. There is some improvement in recognition even on the first few words of the new speaker's input, and this improvement increases markedly over further repetitions of the words.

As with adaptation of speaker-specific templates, it has been found that the application of appropriate compensation factors is essential to the attainment of optimal performance with adaptation. Without compensation, and with randomly ordered input sequences, the results with supervised adaptation of speaker-independent templates show only fairly small and inconsistent improvements at best, and in some cases an overall loss of accuracy, relative to non-adaptive recognition; and the results with unsupervised adaptation are consistently poorer than those without adaptation. With compensation, however, significant improvements are attained by both supervised and unsupervised

adaptation.

The optimal compensation factors (those identified as "K" in table 7.3) for the case with speaker-independent initial templates derived by clustering and averaging, with equal weighting of the speaker-independent template and the new speaker's utterance in the first adaptation (i.e. initial input weight 0.5), have a first factor value (1.15) similar to that occurring in the optimal compensation factors ("h" or "j" in table 6.1) for the case with single-token speaker-specific initial templates, again with equal weighting of initial template and inputs. This indicates that the effect on typical word distance values of the change (occurring at the first adaptation of each template) from speaker-independent cluster average templates to speaker-specific templates, each incorporating one utterance from the current speaker, is similar to the effect of the change from speaker-specific single-utterance templates to speaker-specific two-utterance templates. The increase in the effect on word distance which might be expected in the case with speaker-independent initial templates (relative to that with speaker-specific ones), due to the fact that the initial template was not specific to the current speaker, appears to be balanced by the effect of the fact that in the other case the initial speaker-specific template was derived from only a single utterance and so did not have the benefit of the smoothing effect of averaging (as used in the formation of cluster templates).

The experiments described in this chapter have been limited in that only one set of training utterances, from a fixed set of 50 speakers, has been used to construct the speaker-independent initial templates: only the parameters of the clustering procedure applied to the data have been varied to produce different template sets. The effects of using greater or smaller numbers of training utterances, from the same or different speakers, have not been explored. However, it seems reasonable to expect that the qualitative results obtained as to the

improvement of recognition performance through adaptation would hold true over a wide range of initial template sets.

Another limitation is that only one vocabulary (that of 10 digits) has been considered. This vocabulary is, however, one which is likely to occur, on its own or as part of a larger vocabulary, in many of the possible practical applications for speaker-independent isolated word recognition.

A more serious limitation is that only 30 utterances from each speaker – three repetitions of the vocabulary – were available in the data base for these experiments. Because of this, it has not been possible to explore the longer-term effects of adaptation of initially speaker-independent templates. An exploration of such effects would be of particular interest in the case of unsupervised adaptation, where it is desirable to ensure stability over long input sequences. The problem of instability will be less easily solved by the provision of a retraining facility in many potential applications of speaker-independent word recognition, because these applications involve interaction over telephone lines, where speech is the only means of communication from the user to the system: this makes the specification by the user of the word to be retrained more problematic than in a case (as described in chapter 5) where a keyboard input facility is available. (A procedure for retraining in a system using only speech communication has been devised [225], but it is somewhat more cumbersome than the procedures which can be adopted when other channels of communication are available.)

It would also be of interest to try interactive adaptive recognition using speaker-independent initial templates, rather than performing experiments only on prerecorded input data as was done here. (It was not feasible to use the speaker-independent template sets constructed for these experiments for recognition of input data collected using the interactive recognition system as in chapter 6, because of the differences in the acoustic background conditions and

microphones used, and also (though this could have been overcome) in the LPC analysis applied.) In an interactive recognition task, it would become evident whether the speaker-independent templates corresponded well enough to the current speaker's pronunciations to allow recognition, and hence adaptation, to get started on every word of the vocabulary. If this were not the case, the user would have to retrain the system for the words which failed to be recognised correctly; then the template set in use after retraining would be a mixture of speaker-independent and speaker-specific templates (with perhaps some adapted templates – initially speaker-independent, but now containing speaker-specific information – if other words of the vocabulary had already been recognised).

CHAPTER 8

CONCLUDING DISCUSSION

8: CONCLUDING DISCUSSION

8.1: Review of results

8.1.1: Segmentation and segment representation techniques

The first topic addressed in this research was the comparison of several varieties of time segmentation and segment representation techniques, applied as preprocessing to template-based isolated word recognition with DTW alignment. The experiments and results are described in detail in section 4.2.

Two segmentation techniques were compared: one (linear time segmentation) dividing the endpoint-detected word into a specified number of equal time intervals (without any reference to the acoustic data); and one (trace segmentation) which used a distance measure, applied to each pair of consecutive frame representations, to obtain segments containing equal amounts of acoustic change. In each case, the number of segments per word was fixed, and the segment length (measured in time or in acoustic distance) was adjusted to yield the specified number of segments for each word which was segmented.

Three segment representation techniques were also compared (applicable to segments defined by either of the segmentation techniques): linear interpolation at each segment boundary, selection of the nearest frame vector at each boundary, and averaging of the frame vectors for each segment.

The results showed little overall difference in effectiveness between linear time segmentation and trace segmentation. However, with small numbers of segments per word especially, there was a preference for linear time segmentation in the case of one vocabulary (the digits), and for trace segmentation for the other vocabulary (of mainly disyllabic and polysyllabic words), and the difference between vocabularies in this respect was consistent across the three

speakers. It is difficult to generalise from these results, since they were obtained using only two vocabularies and only three speakers.

The comparison of the different segment representation techniques showed that selection was consistently poorer than interpolation, and that averaging was better than interpolation when the number of segments per word was small, but interpolation was slightly better when the number of segments approached the average number of frames per (unsegmented) word. These results are what might have been predicted from theoretical considerations. Firstly, the technique which makes the use of most information from the unsegmented word is the technique which yields the best recognition accuracy. When the number of segments per word is small, only averaging uses all the original frame vectors, and so it yields the best results. Secondly, there is some benefit from smoothing the acoustic vectors in neighbouring time frames. Selection does not do this at all; interpolation and averaging both result in smoothing, but some of the smoothing effect of averaging is lost when some segments contain only one frame vector each (as happens when the number of segments is more than half the number of frames), and so interpolation becomes better than averaging when this occurs.

The recognition accuracies obtained with large numbers of segments per word (and interpolation) were slightly better than those obtained without any segmentation. Also, accuracies only a little poorer than these were obtained with much smaller numbers of segments per word (and averaging); and moderate accuracies (error rates about 1.5 times or twice those for the optimal case, for the respective vocabularies) were attained using only two segments per word, with averaging.

More significant and reliable results (particularly on the comparison of linear time segmentation and trace segmentation) might have been obtained by using a larger data base, preferably collected from a larger number of speakers (rather than reusing the same small set of data as recognition input with several template sets for each speaker). However, the facilities available at this stage in the project made the collection and processing of data a time-consuming task. It may be reasonably assumed that the main results on segment representation and on numbers of segments per word, which are plausible on theoretical considerations, would be confirmed by any more extensive experiments.

8.1.2: Multiple-stage recognition

The results of the segmentation experiments led to the idea of a multiple-stage recognition system using different numbers of segments at the successive stages. This could improve computational efficiency, by allowing the most unlikely candidate templates to be eliminated at an early stage using a simple comparison with few vectors per word; and it might also improve the overall accuracy by combining the discriminatory powers of different segmentations.

Experiments with a multiple-stage system (described in section 4.4) showed that the average computation per recognition could be reduced by a factor of about 20 in the case of the vocabulary of 10 digits, or 30 in the case of a more confusable 50-word vocabulary, using three stages with appropriate segmentation parameters and template elimination thresholds. The three stages involved representing each word by two, 10 and 30 vectors respectively. The recognition accuracy with the three-stage comparison was similar, and in some cases slightly superior, to that attained using only the third stage. No improvement over the best three-stage performance was attained by using a four-stage com-

parison.

The three-stage structure was adopted for all the subsequent experiments. This facilitated the collection of data in interactive sessions (without subjecting the speakers using the recognition system to long delays between utterance and response), and also allowed experiments to be performed much more efficiently than would have been possible with a one-stage comparison.

One minor drawback of a multiple-stage recognition procedure is that it makes it difficult to define a sensitive recognition quality measure based on correct-word and incorrect-word distances (like the measure R defined in section 4.2.3). The problem lies in pooling statistics of word distance ratios across cases where the recognition decision is taken at different stages. For instance, how much better is a recognition at the first stage, with a best-incorrect/correct word distance ratio slightly greater than the threshold value t_1, than a recognition at the second stage, with a much smaller word distance ratio, in a case where the ratio at the first stage was only slightly below t_1? Because of the difficulty of devising a suitable recognition quality measure for a three-stage system, the results of all the experiments using this system were assessed using only the correctness or incorrectness of each recognition. However, the loss of sensitivity in measuring performance, due to the lack of a measure based on distance values, is greatly outweighed by the smoothing effect of the increased number of experimental trials made possible by the improved efficiency with the three-stage recognition.

In view of the correspondence between distances and (negative) log probabilities in the word model (section 2.5), it might be more appropriate that an additive threshold (or threshold on the difference of word distances) should be used for template elimination after each non-final comparison stage, rather than a multiplicative threshold (or threshold on the word distance ratio). A

multiplicative threshold is, however, convenient in that it is likely to be less necessary to adjust its value whenever the form of the acoustic representation of each frame or segment of speech is changed (e.g. to a vector of 12 linear predictive cepstral coefficients instead of eight mel frequency cepstral coefficients).

A general feature of a multiple-stage decision procedure, with elimination of templates depending on the best word distance at each stage, is that it introduces non-monotonicity into the recognition process. For instance, for a given input utterance, adding more (incorrect) templates to the template set in use can occasionally improve the recognition (changing an incorrect recognition into a correct one), by eliminating at an early stage one of the original incorrect templates which otherwise would cause an error at a later stage – whereas in a one-stage system (with all templates matched in full) adding more incorrect templates could only make the recognition worse.

8.1.3: Template adaptation

The principal focus of the research project was on the adaptation of templates during a recognition session. Experiments were conducted with both speaker-specific and speaker-independent initial templates, using a number of different forms of adaptation (as reported in chapters 6 and 7).

The main conclusion of the adaptation experiments is that adaptation of templates during the recognition process can substantially and reliably improve recognition performance. This is particularly true of supervised adaptation (where there is feedback from the user as to the correctness or incorrectness of each recognition); smaller improvements were observed with unsupervised adaptation. With the system which has been implemented, supervised adaptation does not require an explicit response from the user to each recognition: it is

assumed that the recognition is correct unless the user takes action to correct it. Since the correction of any errors is a necessary part of the user-system interaction in most applications of isolated word recognition (whether or not the templates are being adapted), it should usually be possible in practice to apply supervised adaptation without imposing any extra requirements on the user.

When the initial templates are derived from training utterances by the speaker who is to use the recognition system, the possible benefits of adaptation include the following:-

(1) Templates formed from one or two utterances of each word of the vocabulary can be improved by the incorporation of further utterances. This allows the reliability of multiple-token templates to be attained without the necessity of a lengthy training session before use of the system can begin.

(2) If the difference between the tasks of training and using the system results in a difference in the user's manner of speaking, so that templates formed in a training session are not fully representative of the pronunciations occurring during a recognition session, then adaptation can correct this effect, because the additional utterances which are averaged in to each template are taken from the recognition input.

(3) Adaptation in which the greatest effective weights in the adapted template are given to the most recent utterances used in forming it (as happens with the tracking form of adaptation described in section 5.3, and also with the more sophisticated form proposed in section 6.4) can keep track of gradual changes in the speaker's voice and pronunciations, whether within one session or over a period of many days, and of differences in the voice, steady background noise or manner of pronunciation from session to session.

The experiments conducted with speaker-specific initial templates (chapter 6) measured mainly the first of these three effects. It was found that the recognition error rate occurring with single-utterance initial templates (taken, incidentally, from recognition sessions rather than training sessions, so that effect (2) above could not occur) was reduced typically by a factor of about 3 in the case of the digits vocabulary, or nearly 2 in the case of the more difficult 50-word vocabulary, after a sufficient number of adaptations of each template.

The contribution of effect (2) could be assessed by comparing recognition results obtained on the same input utterances, without adaptation, using templates taken from training sessions and from recognition sessions. To achieve statistical significance, it would be necessary to collect a large number of sets of templates, in training sessions on different occasions, from the speakers who were participate in the recognition sessions to provide the test input. This was not attempted in the research reported here.

As explained in section 6.4, the measurement of effect (3) would require the collection of large amounts of data, to allow the effects of random variations to be overcome, given the restrictions imposed on the use of the data by the necessity of preserving chronological order. The data collection should also ideally be done in a more rigorously controlled manner than was adopted here, with specified intervals between sessions, and with sessions distributed in a principled way across the range of possible times of day. To measure long-term speaker drift effects, it would, of course, be necessary to extend the collection of data from each speaker over a period of several months or preferably years.

When adaptation is applied to speaker-independent initial templates, it has a further potential benefit, in that templates which contain no information about the characteristics of the current speaker (being formed from utterances by a standard set of training speakers) can be made speaker-specific. Experiments

with speaker-independent templates were conducted (as described in chapter 7), to explore the improvements attainable through adaptation to a specific new speaker.

It might have been expected that adaptation would yield greater and more rapid improvements when applied to speaker-independent templates than when applied to initial templates already specific to the current speaker. This expectation was not clearly confirmed by the experiments conducted with speaker-independent initial templates, however. With the best of the sets of speaker-independent templates used in these experiments (set D6), the accuracy attained on the third 10-digit subsequence of a randomly-ordered 30-digit sequence, with prior adaptation on (typically) two repetitions of each digit during the recognition of the preceding 20 digits, was increased by about 3%, from 93% to 96%, relative to the case without adaptation – whereas in the experiments with speaker-specific initial templates (and recognition input from a single session, randomly ordered) the average improvement attained on the third 10-digit subsequence was about 2% (from 96% to 98%). The difference between these two results is not highly significant (as assessed from their respective standard errors: the comparison technique described in the appendix cannot be applied to this comparison since the speaker sets were different); and the proportional reduction in the error rate is similar in the two cases. It is in any case difficult to make a proper comparison of these results with speaker-specific and speaker-independent initial templates, because the recording conditions, the speakers and also the adaptation weighting used were different in the two sets of experiments.

A possible reason for the lack of rapid improvement during the adaptation of speaker-independent templates is that the initial performance on some words of the vocabulary may be very poor, so that it takes several repetitions of a word

before it is recognised correctly and its template is adapted. (The words for which adaptation is slowest, because of this difficulty in getting started, are also, in the nature of the case, the words whose templates are in most need of adaptation.) It still seems reasonable to suppose, however, that, provided that adaptation can get started on each word of the vocabulary, the ultimate improvement on speaker-independent initial templates will be greater than that on speaker-specific ones (even if it takes a larger number of utterances before near-optimal performance is achieved) — because the performance with the unadapted templates is poorer, leaving more room for improvement. (In interactive use of a speaker-independent recognition system, the user will not usually tolerate persistent failure to recognise a particular word correctly, and will prefer to retrain the template for the word concerned, if the facility for retraining is provided. This should result in adequate recognition of the word, using the new (speaker-specific) template, so that adaptation of this template can proceed as in the case with speaker-specific initial training.)

Another possible factor reducing the contrast between the results with speaker-specific and speaker-independent templates is that in these experiments each speaker-specific template was formed (usually) from a single utterance, and was thus liable to be affected by random variability in the speaker's voice and manner of pronunciation and in the acoustic background, whereas the templates in the speaker-independent sets were mostly averaged from several utterances each, which would have a smoothing effect. (This is apparent in the fact (noted in section 7.4) that similar amounts of compensation were required for the reductions in typical word distance due to adaptation when the initial templates were single-token speaker-specific ones and when they were speaker-independent cluster-average templates.)

To obtain the recognition improvements with adaptation discussed above, it was found to be necessary to apply a compensation technique, whereby each word distance was multiplied by a factor depending on the number of times the template had been adapted. Without this compensation, the distance for an adapted incorrect-word template can often be smaller than the distance for an unadapted correct-word template, and this causes recognition errors particularly on the first few repetitions of each word in the input sequence. With correct compensation, the accuracy is significantly improved by the adaptation, even on the first few utterances – as the means and standard errors of improvements (due to supervised adaptation) for the first 10-digit subsequences in tables 6.7 and 7.5-7.7 indicate. Appropriate compensation factors for adaptation of speaker-specific and speaker-independent initial templates (with equal weighting of the initial template and of each input used to adapt it) would appear to be those designated "h" and "K" in tables 6.1 and 7.3 respectively.

Adaptation can be applied not only when the recognition is known to be correct (so as to improve the template's correspondence to the recognised input utterance), but also when it is known to be incorrect (to make the template less like the wrongly recognised input). This can be achieved by using the same weighted averaging procedure as in the adaptation to a correctly recognised input, but with a (small) negative weight on the input instead of a positive one. The use of negative adaptation in cases of misrecognition was found to result in a small increase in the recognition accuracy, on average (as stated in section 6.3.2.4); from the results obtained, the significance of this conclusion is moderate. A further option, when the recognition is incorrect, is verification of the second-best candidate word (and perhaps of the third and subsequent candidates, if the second is incorrect), followed by adaptation (positive or negative) of the second-best-matching template. This was found, in experiments with

speaker-independent initial templates (section 7.3.1), to increase considerably the degree of improvement attained through adaptation after recognition of a given number of inputs. Second-best candidate verification does, however, require additional feedback from the user of the recognition system.

The above-mentioned results are all for supervised adaptation. In the case of speaker-specific initial templates, only slightly poorer long-term results (2.16% improvement, instead of 2.84%, after 250 input utterances) were attained with unsupervised adaptation (using a word distance ratio threshold to define the adaptation condition). For speaker-independent initial templates, however, the improvements resulting from unsupervised adaptation were further below those attained with supervised adaptation. With unsupervised adaptation of speaker-independent templates (tables 7.5-7.7), the maximum improvement on the third 10-digit subsequences ranged from 0.41% (D6) to 1.53% (D2), whereas the maximum improvement with supervised adaptation was always more than twice as great, ranging from 2.65% (D4) to 3.57% (D2). (For comparison, the maximal third-subsequence improvements with speaker-specific initial templates were 1.28% (unsupervised) and 2.03% (supervised), when – as in the speaker-independent templates' case – the test utterances in each trial were all from a single session (table 6.7). In this case, the smaller ratio of supervised to unsupervised adaptation improvements seems unlikely to be due simply to the use of a smaller supervised adaptation weight than in the case with speaker-independent templates (0.25 instead of 0.5), in view of the results in table 6.8; this suggests that it is a genuine effect of the difference between speaker-specific and speaker-independent initial templates, with their corresponding different levels of non-adaptive recognition performance.) This difference between the cases of speaker-independent and speaker-specific initial templates indicates that, for unsupervised adaptation to approach the performance of supervised

adaptation, each word's initial template (or one of its templates, if there are two or more templates per word) has to correspond fairly well to the realisations of the word occurring in the input. (If some words' templates correspond very poorly to the input, the unsupervised adaptation may never get started for these words, since the word distance ratios for correct recognitions of them may not be large enough to satisfy the adaptation condition.) It is difficult, however, given the limited number of utterances per speaker available for the speaker-independent template adaptation experiments, to predict the results that would be obtained with unsupervised adaptation of speaker-independent templates over more extended input sequences.

For unsupervised adaptation, the use of compensation factors is even more important than for supervised adaptation: the results with randomly ordered input (tables 6.7 and 7.5-7.7) show consistent decreases in accuracy, across all input subsequences, for unsupervised adaptation without compensation.

The full range of possible unsupervised adaptation parameter settings was not explored in these experiments. However, the limited results that have been obtained suggest that, to ensure stability in unsupervised adaptation, the weight on each input in the averaging must be made smaller than the weight on the initial template. (Weights in a template:input ratio 4:1 – i.e. "input weight 0.2" – yielded slightly better long-term results than weights in the ratio 3:1 ("input weight 0.25"), as noted in section 6.3.2.5.) It was also found that, with speaker-independent initial templates and no compensation, the loss of accuracy was greater with a lower threshold (1.1 instead of 1.15) in the adaptation condition (which allowed more of the input utterances to be used in the adaptation); but it is difficult to predict from this what the optimal threshold value is likely to be when correct compensation is applied. The optimisation of the threshold and weight values for unsupervised adaptation is an area where some further

research might usefully be done.

A technique – "skewed" adaptation [238,239] – intended to reduce the risk of instability in unsupervised adaptation (with multiple templates per word) was not observed in these experiments to improve the performance attained. It did reduce the losses of accuracy when no compensation was applied, but it also led to reduced improvements, relative to direct adaptation, when appropriate compensation factors were introduced (as seen in tables 7.5-7.7). Skewed adaptation might, however, result in a useful improvement in stability, and also perhaps in some beneficial effects due to the wider spread of the adaptation across the multiple templates, in a comparison on longer sequences of input utterances, if the data were available for such a comparison.

(The skewed adaptation procedure adopted here was not exactly the same as that in [238,239]. There, skewed adaptation was applied to multiple copies of the same initial template for each word, rather than to multiple (different) initial templates as used here; moreover, there were two sets of copies, and the adaptation was applied in the two sets alternately at successive recognitions of a word, subject to an adaptation condition requiring agreement between the recognition decisions within the two sets. With this rather complicated decision and adaptation procedure, stability and an improvement in accuracy were attained in a recognition system whose basic (non-adaptive) performance was quite poor. It would be possible to implement such a procedure for the speaker-specific recognition task (with one training utterance per word) considered in chapter 6; however, this possibility has not been explored here, since reliable improvements were found to be attainable with unsupervised direct adaptation when appropriate compensation factors were applied.)

The use of template adaptation in the recognition system was intended primarily to improve the level of recognition accuracy. However, a side benefit of adaptation was observed during the experiments: given the three-stage structure of the comparison and recognition procedure in this system, the adaptation of the templates leads to a reduction of the average amount of computation required to recognise each word, because fewer templates are retained at the second and third comparison stages when the templates have been improved by adaptation. (This should apply to any other type of hierarchical decision procedure, such as beam searching, provided that the pruning criterion is of the type shown in figure 3.2 (a threshold depending on the minimal distance in the matching so far) – since the improvement of the templates should result in a better separation between correct and incorrect templates' distances.) The absolute and relative reductions in computation per recognition will depend on the size and confusability of the vocabulary, and the number of templates per word. In these experiments, greater reductions were achieved in the case with speaker-independent initial templates (several per word) than in the case with a single speaker-specific template per word. Only in cases with multiple templates was the reduction in the computation for each recognition sufficient to outweigh the extra computation required for the adaptation process. However, in interactive operation of the system, adaptation can still improve the overall speed, since the adaptation processing following a recognition is run in parallel with the digitisation, endpoint detection and acoustic analysis for the next input utterance.

The computation involved in each adaptation operation can be substantially reduced by replacing the DTW alignment of the template and new utterance to be averaged by a linear alignment. This may result in some reduction in the performance improvement attained through the adaptation; this effect appears

(from the results in table 6.4) to be greater for a vocabulary of longer words than for the digits. DTW alignment was used in all the main adaptation experiments described in the preceding chapters.

8.1.4: Endpoint adjustment

One of the main causes of error in isolated word recognition appears to be the inaccurate location of word endpoints. It is difficult to construct an endpoint detection algorithm which will consistently generate accurate endpoints without using any word-specific information.

Improved results can be obtained (at some computational cost) by methods which may take initial endpoint location estimates from a prior endpoint detection algorithm (such as the one incorporated into *del*), but which leave the final decision as to the endpoints effectively used (in the matching of any particular template) until after the alignment and distance computation has been performed. When such a method is applied, the extended DTW algorithm has the task of optimising (by the usual minimum-distance criterion) the start and end, in input time, of the matching of the template, as well as the steps taken in aligning the template with the input frames between those start and end points.

Examples of such techniques were discussed in section 2.3.5 — namely the technique using an extended input interval and repeatable noise or silence frames or one-frame pseudotemplates at the beginning and end of each template [91,93,160], and the edge-free staggered array DP matching algorithm, in which paths may begin and end anywhere on specified lines of gradient -1 in the input-reference plane [38,73]. The first of these is appropriate if weighting scheme (c) [60], in the input direction, is to be adopted; it allows for possible errors in the initial endpoint detection for the input utterance, but not for the

training utterance (or utterances) used to form the template. The second requires the use of a symmetric weighting scheme such as scheme (d); it allows correction of endpoint detection errors in the cases both of the input utterance and of the template.

A DTW algorithm of the former type, with pseudotemplates, was implemented as an option in the system described in chapters 4 and 5; but it could not be used to full advantage in the experiments conducted with this system, because extended endpoints had not been implemented in the prior endpoint detection program employed in the collection of the input utterances, and it was therefore not used in most of the adaptation experiments. It would be of some interest to collect a new data base, with the data for each utterance extended a few frames beyond the detected endpoints, and apply the pseudotemplate technique (with the extended input), and measure the improvements thus attained over recognition without the input extension and the pseudotemplates (in the cases with and without adaptation).

A limitation of an endpoint-adjusting DTW algorithm in the multiple-stage recognition system is that the endpoint adjustment can be applied effectively only if the number of vectors per word is not too small. This is the case in the final stage of the three-stage system (with the parameters adopted, as stated in section 4.4.4.2), and possibly in the second stage, but not in the first stage. (In the few experiments with the three-stage system in which the pseudotemplate option was used (section 6.3.2.4), it was incorporated only in the third stage.)

When the training utterances as well as the input utterances are subject to automatic endpoint detection, an endpoint adjustment technique such that incorporated in the edge-free algorithm of [38,73], with a symmetric weighting scheme, is likely to be better than one such as the pseudotemplate technique implemented here (with an asymmetric weighting scheme), because it allows for

error in the detection of the training utterances' endpoints as well as those of the input utterances. The same remarks about application in a multiple-stage system apply to both forms of endpoint adjustment technique.

Another form of endpoint adjustment [62] which takes into account the results of word matching, without necessarily incorporating any modification into the DTW algorithm, is one in which the initial endpoint detection algorithm may propose several start points for the input word, and, for each template, a separate DTW matching operation is carried out starting from each of these points in input time, and then the start point yielding the best word distance (after appropriate normalisation for word length) is adopted for the matching of that template. (In each matching operation, the ending frame in the input may be fixed, or left to be discovered by the minimum-normalised-distance criterion.) This has the disadvantage that more computation is liable to be required for the DTW matching since multiple DTW operations have to be carried out for each template.

8.1.5: Miscellaneous results and observations

Some refinements to the recognition system were explored, but not adopted for the main series of experiments, because they did not appear to yield any improvement in performance. Details of these are given in sections 5.3 and 5.4. Weighting of the cepstral coefficients according to their standard deviations (as described in [39]), or according to the formula of [40], was found only to decrease the recognition accuracy. The results with a word-specific distance normalisation technique were inconclusive. (A more sophisticated form of distance normalisation might, however, yield improvements in recognition. A possible method of normalisation – which requires a fairly large number of training

tokens of each word for the estimation of probability distributions – has been described by other researchers [98].)

A general finding of the experiments, mentioned in chapter 6, is that, to obtain statistically significant comparisons of adaptation options, it is often necessary to carry out large numbers of trials, using different input utterances, or different random orderings of the input utterances. The use of different orders of the input utterances is appropriate because the result of an adaptive recognition test varies with the order in which the utterances are presented. To measure the statistical significance of a comparison, the analysis described in the appendix, using means and standard error estimates of differences in results on the same data, should be applied.

For realistic evaluation of a speech recognition system, the test utterances should be collected under conditions similar to those in which the system might be used in practice. The use of an interactive procedure, as adopted here, in which the user can see the recognition result on each input utterance before speaking the next word, helps to provide realistic test data.

The user-system interface design and the response time of the interactive recognition system are important for ensuring acceptability to the user, and thus encouraging effective use of the system and obtaining optimal performance with it. If the system is very slow in its responses, or difficult to use because of poor interface design, or if it does not attain an acceptable level of recognition accuracy, then the user may be unwilling to continue to work with the system, or may not attain optimal accuracy while using it. The user-system interaction should be designed so as to encourage consistency of pronunciation, and full realisation of the system's potential, on the part of the user. Helpful features of the interactive recognition system employed in the data collection for this project (described in chapter 5) include the default verification option (which permits

supervised adaptation without the need for explicit verification of each recognition by the user) and the retraining facility (which allows poor performance on particular words, due to noise or errors in template formation, to be improved). The response time of the system was acceptable to users – though several times real time (typically 10s per input utterance) – when the multi-user computer on which it was running was not heavily loaded; but on some occasions, when several other computationally intensive processes were competing for use of the CPU, it became unacceptably slow (and the data collection was then postponed until the load on the machine had decreased). It is difficult, however, to obtain a reliable quantitative assessment of the effects on recognition performance of such factors as response time, background noise, distractions or the user's familiarity with the system, given the large number of possibly relevant variables (including the system's progressive adaptation of its templates as well as the speaker's adaptation to the system), and the fairly small numbers of speakers and sessions in this data base collection exercise.

8.2: Possible extensions of adaptation

The explorations of template adaptation described here have necessarily been limited in their scope in certain respects, and there are several directions in which the adaptation technique might be further extended.

One possible extension is from discrete utterance recognition to connected speech recognition, using either whole-word templates or reference patterns for smaller linguistic units such as syllables, demisyllables, diphones or phonemes. A method for extracting words or other units from connected speech, given an alignment of that speech with a concatenation of existing reference patterns (e.g. isolated-word templates), has been described [149,166,170,177]. Once the

relevant sections of the connected speech have been identified by means of this alignment, they can be used to form new reference patterns which are more representative of the utterances to be recognised – because they are taken from connected speech rather than isolated-word utterances, or because they are specific to the prospective user of the connected speech recognition system (in the case where the initial reference patterns are speaker-independent). Such a method – the segmental *K*-means clustering algorithm – has been successfully applied to the training of a connected word recognition system, using a set of utterances of known strings of words [166,170]. In this case, the initial templates were used only to extract the occurrences of the words from the training data: they were not incorporated into the templates generated following the training procedure. (This is similar to the use of a zero weight on the initial template in some of the experiments with adaptation of speaker-independent templates described in chapter 7.) However, a similar procedure can be employed during the use of a connected word recognition system (in which the utterances are not known in advance, but are recognised using an algorithm as described in section 2.7 – with possible verification of the recognitions by the user), to enhance the initial templates (however these have been formed) by incorporation of more data. Once an occurrence of a word in the input has been located and verified, the weighted averaging of the existing template with the section of input speech can proceed as in the adaptive isolated word recognition system (as described in section 5.3).

A syntax-constrained connected word recognition program, *cwr*, using the one-stage algorithm [148] with beam searching [64,66], and with a template adaptation option, has been implemented, and has been tested on a language domain with a small vocabulary (33 words) and a simple syntax. Using isolated-word templates, without adaptation, 51% sentence recognition accuracy

was obtained. When the templates for the shorter words in the vocabulary were reduced in duration by linear time segmentation and interpolation, to allow for the more rapid pronunciations occurring in connected speech, the accuracy was improved to 90%. With supervised adaptation (optimisation, initial input weight 1.0) of these reduced templates over 92 utterances (three repetitions of the same 36 sentences, with 16 utterances omitted from one of the repetitions, and the utterance order randomised), the overall accuracy was improved to 99.5%, and the accuracy on the last 46 sentence utterances was 100% instead of 89%. Using templates derived by automatic extraction and averaging from words in 16 known sentence utterances (those omitted from the test set), the accuracy on the 92 test utterances was 99% (99% on the last 46) without adaptation, and 99.5% (100%) with adaptation. (These results are averaged over two random orders of the input.) It was found that large compensation factors were required when the initial templates were formed from isolated utterances and the adaptive recognition was performed on connected sentences: the compensation factors used to obtain the results quoted were (1.00 1.25 1.40 1.47 1.52 1.56 1.60).

These connected word recognition experiments used only one (speaker-specific) template for each word in the vocabulary. It would be possible to use multiple templates in adaptive connected word recognition; this would open up the possibility of automatically developing context-specific variants of an initial template during the adaptive recognition.

Another possible extension of the idea of reference pattern adaptation is from templates to (more general) hidden Markov models. An HMM can express information about the variability of words, more effectively than a template, and so the HMM might continue to improve as more input utterances were incorporated into it, beyond the point where near-optimal performance would be reached in adaptive template-based recognition. The main drawbacks of HMMs

(with many specifiable parameters, rather than the relatively few occurring in the special, highly constrained case corresponding to template matching) as the basis of an adaptive recognition system are that more initial training is required than for templates, to achieve an adequate level of accuracy so that the adaptation can get started, and that the improvement with adaptation will tend to be slower than with templates.

Even within the domain of template-based isolated word recognition, there are various possible extensions and refinements of the adaptation by weighted averaging which has been investigated here. One possibility is to average the new input with the best-matching existing template only if the match between the input and that template is close enough, and otherwise to create an additional template (for the same word of the vocabulary) from the input instead [256]. With this form of adaptation, a recognition system which starts out with one template per word can develop into one with several templates per word; the number of templates created will vary from one word of the vocabulary to another, depending on the range of variant pronunciations of the word occurring in the input. (With such a form of adaptation, the development of contextual variant templates for connected word recognition, as suggested above, could perhaps be achieved without the need for multiple initial templates for each word.)

Where adaptation is unsupervised, it may in some circumstances be beneficial to employ some more sophisticated control procedure than the simple imposition of a threshold on the ratio of the best two word distances (as adopted in the system described here), in order to improve the stability of the adaptive system. Such a procedure has been described [238,239], incorporating skewed adaptation and multiple template sets – as mentioned in section 8.1.3 above.

As was mentioned in section 2.3.2, adaptation can be applied in a speech recognition system at the level of units smaller than the word, such as phonemes or the vectors in a vector quantisation codebook. Some of the ideas developed and explored in this thesis for the adaptation of word templates may also be applicable to the adaptation of patterns representing smaller phonetic or acoustic units. In particular, the questions of user interface and verification will be relevant to any system which performs adaptation during recognition sessions; and the idea of compensation factors may well have applications beyond the domain of word-based speech recognition in which it has been applied here.

8.3: Summary

A template-based isolated word recognition system has been implemented, which incorporates a programmable multiple-stage comparison procedure and a range of template adaptation options, and which can operate interactively (with direct speech input) or in batch mode (with previously digitised and analysed data). The multiple-stage comparison makes the recognition computationally efficient, and the adaptation capability allows the system to learn and thus improve its performance as it receives and recognises more utterances.

Improvements in recognition accuracy have been demonstrated to result from template adaptation, both when the initial templates are speaker-specific (formed from single utterances of the words) and when speaker-independent initial templates are used (formed by clustering from utterances by a set of training speakers). The greatest improvements are possible when the adaptation is supervised, though improvements are also attained with unsupervised adaptation. In most practical applications of isolated word recognition, it should be possible to implement supervised adaptation, which need not involve explicit

verification of each recognition by the user. (It is possible to improve the performance further if, when a recognition is incorrect, the second candidate recognition is presented to the user for verification. This could be extended to "full verification", in which the system ascertains from the user the true identity of each input utterance and adapts the appropriate template. However, these options complicate the user-system interface, and will thus not be suitable for all applications.)

In order to attain optimal performance with the adaptive recognition system, it was found to be necessary to apply a compensation technique, whereby the distance obtained in the comparison for each template is adjusted according to the number of times the template has been adapted. Appropriate compensation parameters have been found experimentally for various cases of adaptive recognition.

The main form of adaptation explored has been the "optimisation" formulation, in which the adapted template is an average of the initial template and input utterances in which all the inputs have equal weight (regardless of their order in time). A "tracking" form of adaptation has also been implemented, in which the adapted template for a word at any point in a recognition session depends mainly on the most recent input utterances of the word; this allows for gradual changes in the speaker's voice and pronunciations. However, experiments to evaluate the tracking formulation would require the collection of a large amount of speech data, since the temporal order of the input data must be preserved in the experiments and so a randomisation technique (as applied in the optimisation experiments) is not permissible.

There remains scope for further exploration of a number of possible extensions of reference pattern adaptation in speech recognition, including extensions to connected speech recognition and to hidden Markov models as well as

variations on adaptive template-based isolated word recognition. In general, adaptation of reference patterns is a valuable enhancement to a speech recognition system, especially where the speech encountered as recognition input is expected to differ systematically in some respect from the training speech, or to exhibit a drift over time, or where it is inconvenient to use an extensive initial training procedure.

APPENDIX

STATISTICAL ANALYSIS OF RESULTS

APPENDIX: STATISTICAL ANALYSIS OF RESULTS

A.1: Comparison of two techniques or recognition tasks

On many occasions in speech recognition research, it is of interest to compare the accuracies attained using two different recognition techniques, or under two different sets of conditions. Examples of comparisons of techniques from the preceding chapters include the comparisons of segmentation techniques and of segment representation techniques (in chapter 4), and the comparisons of adaptive and non-adaptive recognition and of different sets of adaptation parameters and compensation factors (in chapters 6 and 7). Comparisons of different conditions include the comparison of results on different vocabularies. An example involving the combination of vocabulary and recognition technique variations occurs in chapter 4, namely the assessment of the observed vocabulary effect on the preference for linear time segmentation or trace segmentation.

In such cases, it is usually possible to test both of the possibilities being compared on data from the same set of speakers — and in many instances (where it is the processing and recognition techniques, rather than the vocabularies or recording conditions, which are being compared), to test them both on the same set of utterances. This allows variations which are irrelevant to the comparison — perhaps due to the characteristics of particular speakers, or even to the peculiarities of individual utterances — to be cancelled out, by the use, in the statistical analysis, of the *differences between corresponding results* for the two possibilities being compared. That is, if the results (usually recognition accuracies) for the K individual utterances, or individual speakers, in the test data base are $a_1, \cdots, a_K$ for the first possibility A, and (respectively) $b_1, \cdots, b_K$ for the second possibility B, then the statistics should be evaluated on the differences

$c_1, \cdots, c_K$ defined by

$$c_k = a_k - b_k. \tag{A.1}$$

Any systematic variation in the results for the individual utterances or speakers, which affects the results for A and for B similarly, is cancelled out by the subtraction.

(It might in some cases be better, depending on the nature of the systematic variation expected to occur, to use error rate ratios $\frac{100\% - a_k}{100\% - b_k}$ instead of the differences c_k. A problem with using ratios is that if any of the error rates $100\% - b_k$ are particularly small then this may result in a long tail of high ratio values, and this tends to reduce the validity of the normal approximation adopted below; in particular, if $b_k = 100\%$ for any k, then the ratio for that value of k is undefined (if $a_k = 100\%$) or infinite. There are various other possible operations which could be applied to the results a_k and b_k to cancel systematic variations; but only the simple case of subtraction is considered here.)

The object of the comparison is to find out whether A is better than B (or whether it is worse) in terms of the recognition accuracies attained. That is, a statistical test is to be applied to distinguish the two hypotheses

$$H_A = \{A \text{ is better than } B\} \tag{A.2}$$

and

$$H_B = \{B \text{ is better than } A\}. \tag{A.3}$$

(The possibility that A and B yield exactly equal performance in general on the recognition task in view is assumed to have probability zero, and may therefore

be neglected.)

Unless there is some strong *a priori* reason to suppose that one of these two hypotheses is more probable than the other, the prior probabilities $P(H_A)$ and $P(H_B)$ may both be set to 0.5. Then the desired output of the statistical test is, by Bayes' theorem,

$$P(H_A \mid a_1, \cdots, a_K, b_1, \cdots, b_K)$$

$$= \frac{0.5P(a_1, \cdots, a_K, b_1, \cdots, b_K \mid H_A)}{0.5P(a_1, \cdots, a_K, b_1, \cdots, b_K \mid H_A) + 0.5P(a_1, \cdots, a_K, b_1, \cdots, b_K \mid H_B)} \quad \text{(A.4a)}$$

$$= \frac{P(c_1, \cdots, c_K \mid H_A)}{P(c_1, \cdots, c_K \mid H_A) + P(c_1, \cdots, c_K \mid H_B)} \quad \text{(A.4b)}$$

on the assumption that only the differences c_k are important for distinguishing H_A and H_B.

Let the difference between the mean recognition accuracies for A and for B (on the theoretically infinite population from which the K samples have been drawn) be r. (This is also equal to the mean of the difference of recognition accuracies between A and B.) If $c_1, \cdots, c_K$ are independent random samples from the population of recognition accuracy differences between A and B, then the minimum-variance unbiased estimate of r given $c_1, \cdots, c_K$ is

$$\bar{c} = \frac{1}{K} \sum_{k=1}^{K} c_K. \quad \text{(A.5)}$$

The standard error of this estimate is

$$\eta = \frac{\sigma}{\sqrt{K}} \quad \text{(A.6)}$$

where σ is the standard deviation of the differences between accuracies with A

and accuracies with B on the population. The population standard deviation σ is estimated by

$$\hat{s} = \left[\frac{1}{K-1}\sum_{k=1}^{K}(c_k - \bar{c})^2\right]^{\frac{1}{2}}. \tag{A.7}$$

The corresponding estimate of the standard error η of $\bar{c}$ as an estimate of r is

$$\hat{e} = \frac{\hat{s}}{\sqrt{K}}. \tag{A.8}$$

It may be assumed that $\bar{c}$, being a sum of independently and identically distributed random variables $\frac{1}{K}c_1, \cdots, \frac{1}{K}c_K$, is approximately normally distributed, with mean r and standard deviation η. This normal approximation becomes more accurate, in general, as K is increased; its accuracy for small values of K depends on the distribution of the individual accuracy differences c_k. Assuming the validity of this normal approximation, the statistic

$$t = \frac{\bar{c} - r}{\hat{e}} \tag{A.9}$$

(for a fixed value of r) has a t distribution with $K-1$ degrees of freedom [261].

The hypothesis H_A (A.2) can be expressed in terms of r as

$$H_A = \{r > 0\}; \tag{A.10}$$

and similarly

$$H_B = \{r < 0\}. \tag{A.11}$$

In computing probabilities of H_A and H_B (posterior probabilities, given the samples $c_1, \cdots, c_K$), r must be treated as a value of a random variable R. Assume that the *a priori* distribution of R is symmetric about 0, and is uniform in the vicinity of 0. This *a priori* distribution may be approximated by the uniform distribution on the interval of real numbers r such that $-w \leq r \leq w$, for some positive real number w. This fulfils the requirement, stated above, that the *a priori* probability of H_A should be 0.5. With this form of *a priori* distribution, the *a posteriori* probability (A.4b) becomes

$$P(H_A \mid c_1, \cdots, c_K) = \frac{\int_0^w \frac{1}{w} f(\frac{\bar{c}-r}{\hat{e}}) dr}{\int_0^w \frac{1}{w} f(\frac{\bar{c}-r}{\hat{e}}) dr + \int_{-w}^0 \frac{1}{w} f(\frac{\bar{c}-r}{\hat{e}}) dr} \tag{A.12a}$$

$$= \frac{\int_0^w f(\frac{\bar{c}-r}{\hat{e}}) dr}{\int_{-w}^w f(\frac{\bar{c}-r}{\hat{e}}) dr}, \tag{A.12b}$$

where f is the probability density function of t (defined by (A.9)). (A factor $P(c_1, \cdots, c_K \mid R=r, t=\frac{\bar{c}-r}{\hat{e}})$, not shown in (A.12), occurs in each integral, but can be cancelled, since it is equal to $P(c_1, \cdots, c_K \mid \bar{c}, \hat{e}, R=r)$, which is independent of r on the assumption that the only dependency between the true mean difference r and the observations $c_1, \cdots, c_K$ is through $\bar{c}$ and $\hat{e}$.) Transforming to integration with respect to t, (A.12b) becomes

$$P(H_A \mid c_1, \cdots, c_K) = \frac{\int_{\frac{\bar{c}-w}{\hat{\epsilon}}}^{\frac{\bar{c}}{\hat{\epsilon}}} \hat{\epsilon} f(t)\,dt}{\int_{\frac{\bar{c}-w}{\hat{\epsilon}}}^{\frac{\bar{c}+w}{\hat{\epsilon}}} \hat{\epsilon} f(t)\,dt} \tag{A.13a}$$

$$= \frac{F(\frac{\bar{c}}{\hat{\epsilon}}) - F(\frac{\bar{c}-w}{\hat{\epsilon}})}{F(\frac{\bar{c}+w}{\hat{\epsilon}}) - F(\frac{\bar{c}-w}{\hat{\epsilon}})} \tag{A.13b}$$

where F is the cumulative distribution function for t. As the width $2w$ of the *a priori* uniform distribution assumed for R is increased, the value of this expression approaches $F(\frac{\bar{c}}{\hat{\epsilon}})$. That is, the confidence in H_A is given by a one-tailed t test, with $K-1$ degrees of freedom, on $\frac{\bar{c}}{\hat{\epsilon}}$.

(For an instance of the use of t tests in evaluation of comparative speech recognition results, see [218]; there the results of the tests (not stated whether one-tailed or two-tailed) are expressed as significances p = (1-confidence), instead of confidences as in this thesis. Cf. [227].)

A.2: Treatment of hierarchical distributions

The procedure proposed in the preceding section assumes the availability of K independent samples $c_1, \cdots, c_K$ from the distribution of differences between results with A and with B.

If there are K independently-collected individual test utterances, then each difference c_k can be taken to be the difference between the (binary-valued) correctnesses of recognition with technique A and with technique B on the kth of these utterances: thus c_k is 0 if the recognitions with A and with B are both correct, or both incorrect, and 1 or -1 if they differ in correctness. Then the samples for which $c_k = 0$ can be ignored (since they carry no information for deciding which of A and B is better), and significance testing can be carried out on the subset of values of k for which $c_k \neq 0$. An example of the comparison procedure for such a case has been given elsewhere [262]. (The approach taken in [262] differs from that taken above, in that a two-tailed test is applied to find a significance level for rejecting the null hypothesis that A and B have indistinguishable performance (and the standard error estimate for the mean of the set of non-zero samples c_k is taken from this null hypothesis, rather than from the observed distribution). However, the same strategy of using the differences c_k between corresponding results a_k and b_k is adopted.)

The assumption of K independent single-utterance samples is appropriate if the K utterances are from speakers selected independently from the population to be modelled.

If the population is, in principle, infinite, then, to satisfy the independence requirement, the utterances should be from K different speakers. This is unlikely to be a practicable way to evaluate a speaker-trained recogniser, since each of the K speakers has to train the system, and the value of K required to obtain significant results with only one test utterance per speaker will generally be very large (typically of the order of several hundreds or thousands). For a speaker-independent recogniser, however, it may be practicable, given a suitable multi-speaker data base. Even so, it will usually be possible to improve the reliability of the results by using several test utterances (for instance, one utterance

of each word in the vocabulary) from each speaker in the data base, rather than just a single one.

The opposite extreme to an infinite population is the case where the population being considered consists of only one speaker. In this case, the K "speakers selected independently from the population" (stipulated above) will all be the same speaker. That is, the differential results on K utterances from a single speaker are acceptable as K independent samples $c_1, \cdots, c_K$, provided that only the comparison of A and B for this particular speaker is of interest. There may be cases where this is so – where a recognition system is to be tuned to the requirements of a particular user – but it is not so if a general-purpose recognition system, to be used by any of a large set of possible speakers, is required.

(In the single-speaker case, if the templates are to be formed from utterances by the target speaker, the question of training becomes problematic. To overcome the effects of peculiarities of particular templates, it may be desirable to use several different template sets during the testing, but then the individual results on different test utterances with the same template set will not be independent samples from the speaker's overall distribution, and so the analysis of section A.1 (or that of [262]) will not be correct for the complete set of single-recognition results in the test – unless a new template set is adopted for each single-recognition trial of A and B. (Also, results on the same test utterance with different template sets will not be independent, and so, to obtain the desired set of independent results, it will be necessary to avoid reusing the same test utterances.) In practice, however, if the vocabulary for which the speaker intends to use the system is the same as the test vocabulary, the aim will usually be to optimise the performance over all choices of system parameters, including the choice of a template set, rather than to distinguish between performances (with different choices A and B of the other system parameters)

averaged over several template sets. Thus any two different template sets will correspond to two cases to be distinguished (an A and a B, in the notation used above), and the testing will use the *differences* between these template sets' results on the same utterances, rather than any results averaged or collated across the template sets. In this case, the recognition performance which is to be optimised is a very specialised one, namely the performance for the specified speaker, using the specified vocabulary, with a fixed set of templates. While this may be a useful thing to optimise for a very specific practical application, its optimisation will not necessarily yield any reliable information of interest for the design of speech recognition systems in general.)

If it is not feasible to use a set of K independent single-recognition differences as the samples $c_1, \cdots, c_K$, then the set of samples $c_1, \cdots, c_K$ will have to be derived in some way from a set of (more than K) single-recognition results which are not all independent. In particular, to obtain results applicable to a theoretically-infinite population of speakers, it will usually be necessary to conduct the testing on multiple recognition results for each of a limited set of speakers.

In general, the optimal estimate of the mean difference (r) in recognition accuracy between A and B, based on independent results (accuracy differences) $c_1, \cdots, c_K$ for K different speakers, is

$$\hat{r} = \frac{1}{\sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2}} \sum_{k=1}^{K} \frac{c_k}{\eta_k^2 + \theta^2}, \tag{A.14}$$

where η_k is the standard error of c_k as an estimate of the mean r_k for the kth speaker, and θ is the standard deviation of the distribution of single-speaker means r_k about r. (This does not necessarily hold true if the standard errors η_k

are correlated with the means r_k, since then the weighting of the samples c_k in (A.14) will introduce a bias towards those values of r_k which correspond to small values of η_k, so that the statistic $\hat{r}$, while still having minimal variance over all weighted sums of the samples c_k, may not be an unbiased estimate of r. In this case, a bias term must be introduced. However, if the variances η_k are fixed by some criterion which is independent of the speaker means r_k — for instance, if they are determined by the numbers of utterances collected from the speakers, and these are arbitrarily fixed without reference to any characteristics of the speakers — then (A.14) will give an unbiased estimate of r.) The standard error of $\hat{r}$ is

$$\eta = \left[\frac{1}{\left[\sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2} \right]^2} E\left[\sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2} (c_k - r)^2 \right] \right]^{\frac{1}{2}} \tag{A.15a}$$

(since the samples c_k are independent)

$$= \left[\frac{1}{\left[\sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2} \right]^2} \sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2} \left[E[(c_k - r_k)^2] + E[(r_k - r)^2] \right] \right]^{\frac{1}{2}} \tag{A.15b}$$

$$= \left[\frac{1}{\left[\sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2} \right]^2} \sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2} \left[\eta_k^2 + \theta^2 \right] \right]^{\frac{1}{2}} \tag{A.15c}$$

(using the definitions of η_k and θ)

$$= \frac{1}{\left[\sum_{k=1}^{K} \frac{1}{\eta_k^2 + \theta^2} \right]^{\frac{1}{2}}}. \tag{A.15d}$$

In the case where each single-speaker difference c_k is an average over n recognitions (for some constant n), and the structure of the recognition tests is the same

for all speakers (using the same number of template sets, the same amount of input data, the same randomisation techniques, etc.), it is plausible to use an approximation to (A.14) in which the standard errors η_k are taken to be the same for all speaker indices k. In this case

$$\hat{r} = \frac{1}{K}\sum_{k=1}^{K} c_k, \tag{A.16}$$

and

$$\eta = \frac{\sigma}{\sqrt{K}}, \tag{A.17}$$

where

$$\sigma^2 = \eta_k^2 + \theta^2 \tag{A.18}$$

(which is the same for all values of k). Thus, when equal standard errors η_k are assumed for the single-speaker mean results, the estimate of r obtained from a hierarchical distribution of results (grouped by speaker) is simply $\bar{c}$ (as in (A.5)), where each sample c_k is the mean accuracy difference for one (the kth) of the K speakers. Also the estimated standard error $\hat{e}$ is given by (A.8) – since

$$E\left[\hat{s}^2\right] = \frac{1}{K-1}\sum_{k=1}^{K} E\left[((c_k - r) - (\bar{c} - r))^2\right] \tag{A.19a}$$

$$= \frac{1}{K-1}\sum_{k=1}^{K} E\left[\left[\frac{K-1}{K}(c_k - r) - \sum_{j \neq k}\frac{c_j - r}{K}\right]^2\right] \tag{A.19b}$$

$$= \frac{1}{K-1}\sum_{k=1}^{K}\left[\frac{(K-1)^2}{K^2}E\left[(c_k - r)^2\right] + \frac{1}{K^2}\sum_{j \neq k} E\left[(c_j - r)^2\right]\right] \tag{A.19c}$$

(since the random variables $c_j - r$ and $c_k - r$ are independent with mean 0)

$$= \frac{1}{K-1}\sum_{k=1}^{K}\left[\frac{(K-1)^2}{K^2}(\eta_k^2+\theta^2)+\frac{1}{K^2}\sum_{j\neq k}(\eta_j^2+\theta^2)\right] \tag{A.19d}$$

(by substitutions as in (A.15a-c))

$$= \frac{1}{K-1}\sum_{k=1}^{K}\left[\frac{(K-1)^2}{K^2}\sigma^2+\frac{K-1}{K^2}\sigma^2\right] \tag{A.19e}$$

$$= \sigma^2. \tag{A.19f}$$

The details of (A.14-A.19) are unnecessary if the single-speaker average differences c_k are treated simply as independent samples from the same distribution, as in section A.1. However, the above analysis demonstrates how this simple case relates to the more general case with different theoretical single-speaker means r_k and standard errors η_k, and this may be of some interest. The important point to note is that the quantities c_k used in the mean and standard error estimation must be *independent* samples from the distribution being studied, and therefore if the individual recognition results (differences in single-utterance recognition correctness) are from a hierarchy of distributions (as in the case with a number of results for each of a set of test speakers) then these individual results must be grouped together (e.g. by averaging the results for each speaker) to form a set of independent samples from a distribution at the most general level of the hierarchy, before the mean, the standard error estimate and the corresponding confidence are computed.

REFERENCES

The reference list is arranged by subject categories (as listed in the table of contents, on page viii); within each category the references are given in chronological order. The order of the subject categories corresponds roughly to the order in which the topics are treated in the text of the thesis. At the end of each category are listed any references which have been placed in other categories but are also relevant to the topic of this category. An alphabetical index of authors is provided after the reference list.

R.1: Principles of pattern matching and speech recognition

[1] R. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[2] T.K. Vintsyuk, "Speech Discrimination by Dynamic Programming", *Kibernetika (Cybernetics)*, vol.4, pp.81-88, January-February 1968.

[3] T.K. Vintsyuk, "Element-wise Recognition of Continuous Speech Consisting of Words of a Given Vocabulary", *Kibernetika (Cybernetics)*, vol.7, no.2, pp.361-372, 1971.

[4] S.R. Hyde, "Automatic Speech Recognition: A Critical Survey and Discussion of the Literature", pp.399-438 in E.E. David and P.B. Denes (eds), *Human Communication: A Unified View*, McGraw Hill, 1972.

[5] G.D. Forney, "The Viterbi Algorithm", *Proc. IEEE*, vol.61, pp.268-278, March 1973.

[6] F. Jelinek, L.R. Bahl and R.L. Mercer, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech", *IEEE Trans. Information Theory*, vol.IT-21, pp.250-256, May 1975.

[7] L.R. Bahl and F. Jelinek, "Decoding for Channels with Insertions, Deletions, and Substitutions with Applications to Speech Recognition", *IEEE Trans. Information Theory*, vol.IT-21, pp.404-411, July 1975.

[8] L.R. Bahl, F. Jelinek and R.L. Mercer, "A Maximum Likelihood Approach to Continuous Speech Recognition", *IEEE Trans. Pattern Anal. and Machine Intelligence*, vol.PAMI-5, pp.179-190, July 1975.

[9] D.R. Reddy, "Speech Recognition by Machine: A Review", *Proc. IEEE*, vol.64, pp.501-531, April 1976.

[10] F. Jelinek, "Continuous Speech Recognition by Statistical Methods", *Proc. IEEE*, vol.64, pp.532-556, April 1976.

[11] D.H. Klatt, "Review of the ARPA Speech Understanding Project", *J. Acoust. Soc. Amer.*, vol.62, pp.1345-1366, December 1977.

[12] L.R. Rabiner and S.E. Levinson, "Isolated and Connected Word Recognition – Theory and Selected Applications", *IEEE Trans. Communications*, vol.COM-29, pp.621-659, May 1981.

[13] J.S. Bridle, "Stochastic Models and Template Matching: Some Important Relationships Between Two Apparently Different Techniques for Automatic Speech Recognition", presented at Inst. of Acoust. Autumn Conf., 1984.

[14] B.-H. Juang, "On the Hidden Markov Model and Dynamic Time Warping for Speech Recognition – A Unified View", *AT&T Bell Laboratories Tech. J.*, vol.63, pp.1213-1243, September 1984.

[15] J. Vaissière, "Speech Recognition: A Tutorial", chapter 8 (pp.191-242) in F. Fallside and W.A. Woods (eds), *Computer Speech Processing*, Prentice/Hall International, 1985.

[16] S.E. Levinson, "A Unified Theory of Composite Pattern Analysis for Automatic Speech Recognition", chapter 9 (pp.243-275) in F. Fallside and W.A. Woods (eds), *Computer Speech Processing*, Prentice/Hall International, 1985.

[17] F.R. McInnes and M.A. Jack, "Automatic Speech Recognition using Word Reference Patterns", chapter 1 (pp.1-68) in M.A. Jack and J. Laver (eds), *Aspects of Speech Technology*, Edinburgh University Press, 1988.

R.2: Acoustic representations and distance measures

[18] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-23, pp.67-72, February 1975.

[19] G.M. White and R.B. Neely, "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering, and Dynamic Programming", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-24, pp.183-188, April 1976.

[20] J.D. Markel and A.H. Gray, *Linear Prediction of Speech*, Springer-Verlag, 1976.

[21] A.H. Gray and J.D. Markel, "Distance Measures for Speech Processing", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-24, pp.380-391, October 1976.

[22] J.M. Tribolet, L.R. Rabiner and M.M. Sondhi, "Statistical Properties of an LPC Distance Measure", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-27, pp.550-558, October 1979.

[23] S.B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", *IEEE Trans. Acoust., Speech, and Signal Process*, vol.ASSP-28, pp.357-366, August 1980.

[24] P. de Souza and P.J. Thomson, "LPC Distance Measures and Statistical Tests with Particular Reference to the Likelihood Ratio", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-30, pp.304-315, April 1982.

[25] K. Elenius and M. Blomberg, "Effects of Emphasizing Transitional or Stationary Parts of the Speech Signal in a Discrete Utterance Recognition System", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.535-538.

[26] P.K. Rajasekaran and G.R. Doddington, "Microcomputer Implementable Low Cost Speaker-Independent Word Recognition" *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.753-756.

[27] C.A. Olano, "An Investigation of Spectral Match Statistics Using a Phonemically Marked Data Base", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.773-776.

[28] B.A. Dautrich, L.R. Rabiner and T.B. Martin, "On the Use of Filter Bank Features for Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1057-1060.

[29] G. Neben, R.J. McAulay and C.J. Weinstein, "Experiments in Isolated Word Recognition Using Noisy Speech", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1156-1159.

[30] B.A. Dautrich, L.R. Rabiner and T.B. Martin, "The Effects of Selected Signal Processing Techniques on the Performance of a Filter-Bank-Based Isolated Word Recognizer", *Bell System Tech. J.*, vol.62, pp.1311-1336, May-June 1983.

[31] B.A. Dautrich, L.R. Rabiner and T.B. Martin, "On the Effects of Varying Filter Bank Parameters on Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-31, pp.793-806, August 1983.

[32] B. Kammerer, W. Küpper and H. Lagger, "Special Feature Vector Coding and Appropriate Distance Definition Developed for a Speech Recognition

System", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 17.4).

[33] M. Blomberg, R. Carlson, K. Elenius and B. Granström, "Auditory Models in Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 17.9).

[34] Y. Dologlou and J.M. Dolmazon, "Comparison of a Model of the Peripheral Auditory System and L.P.C. Analysis in a Speech Recognition System", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 17.10).

[35] J.S. Bridle, K.M. Ponting, M.D. Brown and A.W. Borrett, "A Noise Compensating Spectrum Distance Measure Applied to Automatic Speech Recognition", *Proc. Inst. of Acoust.*, vol.6, part 4, pp.307-314, 1984.

[36] N. Nocerino, F.K. Soong, L.R. Rabiner and D.H. Klatt, "Comparative Study of Several Distortion Measures for Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.25-28.

[37] M.J. Hunt, "A Robust Formant-Based Speech Spectrum Comparison Measure", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1117-1120.

[38] S. Furui, "Speaker-Independent Isolated Word Recognition Using Dynamic Features of Speech Spectrum", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-34, pp.52-59, February 1986.

[39] Y. Tohkura, "A Weighted Cepstral Distance Measure for Speech Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.761-764.

[40] B.-H. Juang, L.R. Rabiner and J.G. Wilpon, "On the Use of Bandpass Liftering in Speech Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.765-768.

[41] M.J. Hunt and C. Lefebvre, "Speech Recognition Using a Cochlear Model", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1979-1982.

[42] S. Furui, "Speaker-Independent Isolated Word Recognition Based on Emphasized Spectral Dynamics", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1991-1994.

[43] M. Blomberg and K. Elenius, "Nonlinear Frequency Warp for Speech Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2631-2634.

[44] E.L. Bocchieri and G.R. Doddington, "Speaker Independent Digit Recognition with Reference Frame-Specific Distance Measures", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*,

April 1986, pp.2699-2702.

[45] E.L. Bocchieri and G.R. Doddington, "Frame-Specific Statistical Features for Speaker Independent Speech Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-34, pp.755-764, August 1986.

(See also [14,48,58,134,160].)

R.3: Isolated word recognition using word templates and DTW

[46] L.R. Rabiner and M.R. Sambur, "An Algorithm for Determining the Endpoints of Isolated Utterances", *Bell System Tech. J.*, vol.54, pp.297-315, February 1975.

[47] V.N. Gupta, J.K. Bryan and J.N. Gowdy, "A Speaker-Independent Speech-Recognition System Based on Linear Prediction", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-26, pp.27-33, February 1978.

[48] L.R. Rabiner, "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-26, pp.34-42, February 1978.

[49] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-26, pp.43-49, February 1978.

[50] Y. Niimi, "A Method for Forming Universal Reference Patterns in an Isolated Word Recognition System", *Proc. 4th Int. Joint Conf. Pattern Recognition*, 1978.

[51] L.R. Rabiner, A.E. Rosenberg and S.E. Levinson, "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-26, pp.575-582, December 1978.

[52] C.C. Tappert and S.K. Das, "Memory and Time Improvements in a Dynamic Programming Algorithm for Matching Speech Patterns", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-26, pp.583-586, December 1978.

[53] S.E. Levinson, L.R. Rabiner, A.E. Rosenberg and J.G. Wilpon, "Interactive Clustering Techniques for Selecting Speaker-Independent Reference Templates for Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-27, pp.134-141, April 1979.

[54] L.R. Rabiner, S.E. Levinson, A.E. Rosenberg and J.G. Wilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques",

IEEE Trans. Acoust., Speech, and Signal Process., vol.ASSP-27, pp.336-349, August 1979.

[55] L.R. Rabiner and J.G. Wilpon, "Speaker-Independent Isolated Word Recognition for a Moderate Size (54 Word) Vocabulary", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-27, pp.583-587, December 1979.

[56] L.R. Rabiner and J.G. Wilpon, "Application of Clustering Techniques to Speaker-Trained Isolated Word Recognition", *Bell System Tech. J.*, vol.58, pp.2217-2233, December 1979.

[57] H.F. Silverman and N. Rex Dixon, "State Constrained Dynamic Programming (SCDP) for Discrete Utterance Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1980, pp.169-172.

[58] S.K. Das, "Some Experiments in Discrete Utterance Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1980, pp.178-181 (later edition published in *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-30, pp.535-544, August 1982).

[59] J.R. Welch and S.C. Oxenberg, "Reduction of Minimum Word-Boundary Gap Lengths in Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1980, pp.190-193.

[60] C. Myers, L.R. Rabiner and A.E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-28, pp.623-635, December 1980.

[61] L.R. Rabiner and J.G. Wilpon, "A Two-Pass Pattern-Recognition Approach to Isolated Word Recognition", *Bell System Tech. J.*, vol.60, pp.739-766, May-June 1981.

[62] L.F. Lamel, L.R. Rabiner, A.E. Rosenberg and J.G. Wilpon, "An Improved Endpoint Detector for Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-29, pp.777-785, August 1981.

[63] L.R. Rabiner, "Notes on Some Factors Affecting Performance of Dynamic Time Warping Algorithms for Isolated Word Recognition", *Bell System Tech. J.*, vol.61, pp.363-373, March 1982.

[64] R. Bisiani and A. Waibel, "Performance Trade-offs in Search Techniques for Isolated Word Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.570-573.

[65] M. Okochi and T. Sakai, "Trapezoidal DP Matching with Time Reversibility", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1239-1242.

[66] K. Greer, B. Lowerre and L. Wilcox, "Acoustic Pattern Matching and Beam Searching", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1251-1254.

[67] M.K. Brown and L.R. Rabiner, "Dynamic Time Warping for Isolated Word Recognition Based on Ordered Graph Searching Techniques", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1255-1258.

[68] K.K. Paliwal, A. Agarwal and S.S. Sinha, "A Modification over Sakoe and Chiba's Dynamic Time Warping Algorithm for Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1259-1261.

[69] R.W. Brown, "Segmentation for Data Reduction in Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1262-1265.

[70] R.K. Moore, M.J. Russell and M.J. Tomlinson, "Locally Constrained Dynamic Programming in Automatic Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1270-1273.

[71] C.A. Vickroy, H.F. Silverman and N.R. Dixon, "Study of Human and Machine Discrete Utterance Recognition (DUR)", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.2022-2025.

[72] G.F. Chollet and C. Gagnoulet, "On the Evaluation of Speech Recognisers and Data Bases Using a Reference System", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.2026-2029.

[73] K. Shikano and K. Aikawa, "Staggered Array DP Matching", *ASJ Trans. Comm. Speech Research*, S82-15, 1982, pp.113-120 (in Japanese: English translation available from AT&T Bell Laboratories Libraries and Information Systems Center).

[74] M.K. Brown and L.R. Rabiner, "An Adaptive, Ordered, Graph Search Technique for Dynamic Time Warping for Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-30, pp.535-544, August 1982.

[75] J.M. Tribolet, L.R. Rabiner and J.G. Wilpon, "An Improved Model for Isolated Word Recognition", *Bell System Tech. J.*, vol.61, pp.2289-2312, November 1982.

[76] M.H. Kuhn and H.H. Tomaschewski, "Improvements in Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-31, pp.157-167, February 1983.

[77] A. Rollins and J. Wiesen, "Speech Recognition and Noise", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.523-526.

[78] N. Sugamura, K. Shikano and S. Furui, "Isolated Word Recognition Using Phoneme-Like Templates", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.723-736.

[79] L. Niles, H.F. Silverman and N.R. Dixon, "A Comparison of Three Feature Vector Clustering Procedures in a Speech Recognition Paradigm", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.765-768.

[80] R. Pieraccini and R. Billi, "Experimental Comparison Among Data Compression Techniques in Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1025-1028.

[81] J.L. Gauvain, J. Mariani and J.S. Lienard, "On the Use of Time Compression for Word-Based Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1029-1032.

[82] C.-K. Chuang and S.W. Chan, "Speech Recognition Using Variable Frame Rate Coding", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1033-1036.

[83] M.J. Russell, R.K. Moore and M.J. Tomlinson, "Some Techniques for Incorporating Local Timescale Variability Information into a Dynamic Time-Warping Algorithm for Automatic Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1037-1040.

[84] R.K. Moore, M.J. Russell and M.J. Tomlinson, "The Discriminative Network: A Mechanism for Focusing Recognition in Whole-Word Pattern Matching", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1041-1044.

[85] L. Wilcox, B. Lowerre and M. Kahn, "Use of A Priori Knowledge of Vocabulary for Real Time Discrete Utterance Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1045-1048.

[86] R. Zelinski and F. Class, "A Learning Procedure for Speaker-Dependent Word Recognition Systems Based on Sequential Processing of Input Tokens", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1053-1056.

[87] J.G. Wilpon, L.R. Rabiner and T. Martin, "An Improved Word-Detection Algorithm for Telephone-Quality Speech Incorporating Both Syntactic and Semantic Constraints", *AT&T Bell Laboratories Tech. J.*, vol.63, pp.479-498, March 1984.

[88] Y.J. Liu, "On Creating Averaging Templates", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 9.1).

[89] V.N. Gupta, M. Lennig and P. Mermelstein, "Decision Rules for Speaker-Independent Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 9.2).

[90] A.E. Rosenberg and K.L Shipley, "Evaluation of an Isolated Word Recognizer in Talker-Dependent and Talker-Independent Modes Using a Large Telephone-Band Data Base", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 9.5).

[91] S. Haltsonen, "An Endpoint Relaxation Method for Dynamic Time Warping Algorithms", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 9.8).

[92] J.-S. Lienard and F.K. Soong, "On the Use of Transient Information in Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 17.3).

[93] S. Haltsonen, "Improved Dynamic Time Warping Methods for Discrete Utterance Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.449-450, June 1985.

[94] J.G. Wilpon and L.R. Rabiner, "A Modified K-Means Clustering Algorithm for Use in Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.587-594, June 1985.

[95] S. Haltsonen, "Recognition of Isolated-Word Sentences From a Large Vocabulary Using Dynamic Time Warping Methods", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.1026-1027, August 1985.

[96] M.R. Taylor, "Isolated word recognition based on distinctive feature extraction and dynamic time warping", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986, pp.8-14.

[97] F.R. McInnes, M.A. Jack and J. Laver, "Comparative study of time segmentation and segment representation techniques in a DTW-based word recogniser", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986, pp.21-26.

[98] C.J. Clotworthy and F.J. Smith, "Spoken word variation and probability estimation", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986, pp.27-30.

[99] T. Nomura and R. Nakatsu, "Speaker-Independent Isolated Word Recognition for Telephone Voice Using Phoneme-like Templates", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2687-2690.

[100] A. Mokeddem, H. Hügli and F. Pellandini, "New Clustering Algorithms Applied to Speaker Independent Isolated Word Recognition", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2691-2694.

[101] M.R. Taylor, "Comparative Isolated Word Recognition Experiments", *Proc. Inst. of Acoust.*, vol.8, part 7, pp.265-273, 1986.

[102] N. Yalabik and A. Mansur, "Using Condensed Nearest Neighbor Rule for Speaker Independent Word Recognition", *Proc. European Conf. on Speech Technology*, September 1987, vol.2, pp.276-279.

(See also [15,16,18,23,30,48,89,109,117,119,160,237,242].)

R.4: Isolated word recognition using hidden Markov models

[103] L.R. Rabiner, S.E. Levinson and M.M. Sondhi, "On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition", *Bell System Tech. J.*, vol.62, pp.1075-1105, April 1983.

[104] S.E. Levinson, L.R. Rabiner and M.M. Sondhi, "Speaker Independent Isolated Digit Recognition Using Hidden Markov Models", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.1049-1052.

[105] K. Sugawara, M. Nishimura, K. Toshioka, M. Okochi and T. Kaneko, "Isolated Word Recognition Using Hidden Markov Models", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1-4.

[106] M.J. Russell and R.K. Moore, "Explicit Modelling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.5-8.

[107] B.-H. Juang, L.R. Rabiner, S.E. Levinson and M.M. Sondhi, "Recent Developments in the Application of Hidden Markov Models to Speaker-Independent Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.9-12.

[108] D.B. Paul, "Training of HMM Recognizers by Simulated Annealing", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.13-16.

[109] M.J. Russell, "Maximum Likelihood Hidden Semi-Markov Model Parameter Estimation for Automatic Speech Recognition", RSRE Memorandum 3837, July 1985.

[110] L.R. Rabiner, B.-H. Juang, S.E. Levinson and M.M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models With Continuous Mixture Densities", *AT&T Tech. J.*, vol.64, pp.1211-1234, July-August 1985.

[111] Y. Kamp, "State Reduction in Hidden Markov Chains Used for Speech Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.1138-1145, October 1985.

[112] B.-H. Juang and L.R. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speech Signals", *IEEE Trans. Acoust., Speech, and Signal*

Process., vol.ASSP-33, pp.1404-1413, December 1985.

[113] B.-H. Juang and L.R. Rabiner, "Mixture Autoregressive Hidden Markov Models for Speaker Independent Isolated Word Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.41-44.

[114] S. Soudoplatoff, "Markov Modeling of Continuous Parameters in Speech Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.45-48.

[115] L.R. Bahl, P.F. Brown, P.V. de Souza and R.L. Mercer, Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.49-52.

[116] A.B. Poritz and A.G. Richter, "On Hidden Markov Models in Isolated Word Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.705-708.

[117] M.J. Russell and A.E. Cook, "Experiments in Speaker-Dependent Isolated Digit Recognition Using Hidden Markov Models", *Proc. Inst. of Acoust.*, vol.8, part 7, pp.291-298, 1986.

[118] A.E. Cook and M.J. Russell, "Improved Duration Modelling in Hidden Markov Models Using Series-Parallel Configurations of States", *Proc. Inst. of Acoust.*, vol.8, part 7, pp.299-306, 1986.

[119] M.J. Russell and A.E. Cook, "Experimental Evaluation of Duration Modelling Techniques for Automatic Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, April 1987, pp.2376-2379.

[120] X.D. Huang and M.A. Jack, "Hidden Markov modelling of speech based on a semi-continuous model", *Electronics Letters*, vol.24, no.1, pp.6-7, January 1988.

[121] X.D. Huang and M.A. Jack, "Performance comparison between semicontinuous and discrete hidden Markov models of speech", *Electronics Letters*, vol.24, no.3, pp.149-150, February 1988.

(See also [6,7,8,10,14,15,171,175].)

R.5: Word-based speech recognition without dynamic programming

[122] Y. Nara, K. Iwata, Y. Kijima, S. Kimura, S. Sasaki and J. Tanahashi, "Large-Vocabulary Spoken Word Recognition Using Simplified Time-Warping Patterns", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1266-1269.

[123] D.K. Burton and J.E. Shore, "Speaker-Dependent Isolated Word Recognition Using Speaker-Independent Vector Quantization Codebooks Augmented with Speaker-Specific Data", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.440-443, April 1985.

[124] D.K. Burton, J.E. Shore and J.T. Buck, "Isolated Word Speech Recognition Using Multisection Vector Quantization Codebooks", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.837-849, August 1985.

[125] O. Watanuki and T. Kaneko, "Speaker-Independent Isolated Word Recognition Using Label Histograms", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2679-2682.

R.6: Use of constraints on isolated word sequences

[126] S.E. Levinson, A.E. Rosenberg and J.L. Flanagan, "Evaluation of a Word Recognition System Using Syntax Analysis", *Bell System Tech. J.*, vol.57, pp.1619-1626, May-June 1978.

[127] S.E. Levinson, "The Effects of Syntactic Analysis on Word Recognition Accuracy", *Bell System Tech. J.*, vol.57, pp.1627-1644, May-June 1978.

[128] A.E. Rosenberg and C.E. Schmidt, "Automatic Recognition of Spoken Spelled Names for Obtaining Directory Listings", *Bell System Tech. J.*, vol.58, pp.1797-1823, October 1979.

[129] S.E. Levinson and K.L. Shipley, "A Conversational-Mode Airline Information and Reservation System Using Speech Input and Output", *Bell System Tech. J.*, vol.59, pp.119-137, January 1980.

(See also [12,15].)

R.7: Word spotting

[130] J.S. Bridle, "An Efficient Elastic-Template Method for Detecting Given Words in Running Speech", *British Acoust. Soc. Meeting*, April 1973, pp.1-4.

[131] J.S. Bridle and N.C. Sedgwick, "A Method of Segmenting Acoustic Patterns, with Applications to Automatic Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1977, pp.656-659.

[132] R.W. Christiansen and C.K. Rushforth, "Detecting and Locating Key Words in Continuous Speech Using Linear Predictive Coding", *IEEE*

Trans. Acoust., Speech, and Signal Process., vol.ASSP-25, pp.361-367, October 1977.

[133] N.C. Sedgwick, "Automatic Speech Recognition" (MTech dissertation for Brunel University), CAP Scientific Ltd, November 1979.

[134] R.E. Wohlford, A.R. Smith and M.R. Sambur, "The Enhancement of Wordspotting Techniques", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1980, pp.209-212.

[135] D.P. McCullough, "Secondary Testing Techniques for Word Recognition in Continuous Speech", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.300-303.

[136] T. Kawabata and M. Kohda, "Word Spotting Taking Account of Duration Change Characteristics for Stable and Transient Parts of Speech", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2307-2310.

(See also [27,139,142,147].)

R.8: Connected word recognition

[137] C. Cook, "Word Verification in a Speech Understanding System", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, April 1976, pp.553-556.

[138] H. Sakoe, "Two-Level DP-Matching – A Dynamic Programming-Based Pattern Matching Algorithm for Connected Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-27, pp.588-595, December 1979.

[139] C.S. Myers, L.R. Rabiner and A.E. Rosenberg, "An Investigation of the Use of Dynamic Time Warping for Word Spotting and Connected Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1980, pp.173-177.

[140] L.R. Rabiner and C.E. Schmidt, "A Connected Digit Recognizer Based on Dynamic Time Warping and Isolated Digit Templates", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1980, pp.194-198.

[141] L.R. Rabiner and C.E. Schmidt, "Application of Dynamic Time Warping to Connected Digit Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-28, pp.377-388, August 1980.

[142] C.S. Myers, L.R. Rabiner and A.E. Rosenberg, "On the Use of Dynamic Time Warping for Word Spotting and Connected Word Recognition", *Bell System Tech. J.*, vol.60, pp.303-325, March 1981.

[143] C.S. Myers and L.R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-29, pp.284-297, April 1981.

[144] C.S. Myers and L.R. Rabiner, "Connected Digit Recognition Using a Level-Building DTW Algorithm", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-29, pp.351-363, June 1981.

[145] C.S. Myers and L.R. Rabiner, "A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition", *Bell System Tech. J.*, vol.60, pp.1389-1409, September 1981.

[146] C. Gagnoulet and M. Couvrat, "Seraphine: a Connected Word Speech Recognition System", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.887-890.

[147] J.-L. Gauvain and J. Mariani, "A Method for Connected Word Recognition and Word Spotting on a Microprocessor", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.891-894.

[148] J.S. Bridle, M.D. Brown and R.M. Chamberlain, "An Algorithm for Connected Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.899-902.

[149] L.R. Rabiner, A. Bergh and J.G. Wilpon, "An Embedded Word Training Procedure for Connected Digit Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1621-1624.

[150] C.S. Myers and S.E. Levinson, "Speaker Independent Connected Word Recognition Using a Syntax-Directed Dynamic Programming Procedure", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-30, pp.561-565, August 1982.

[151] J.S. Bridle, M.D. Brown and R.M. Chamberlain, "Continuous connected word recognition using whole word templates", *The Radio and ElectronicEngineer*, vol.53, pp.167-175, April 1983.

[152] H. Ney, "Experiments in Connected Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.288-291.

[153] S. Nakagawa, "A Connected Spoken Word Recognition Method by O(n) Dynamic Programming Pattern Matching Algorithm", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.296-299.

[154] P.F. Brown, C.-H. Lee and J.C. Spohrer, "Bayesian Adaptation in Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.761-764.

[155] R.M. Chamberlain and J.S. Bridle, "ZIP: A Dynamic Programming Algorithm for Time-Aligning Two Indefinitely Long Utterances", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.816-819.

[156] H. Bourlard, C.J. Wellekens and H. Ney, "Connected Digit Recognition Using Vector Quantization", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 26.10).

[157] B.P. Landell, J.A. Naylor and R.E. Wohlford, "Effect of Vector Quantization on a Continuous Speech Recognition System", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 26.11).

[158] L.R. Rabiner, J.G. Wilpon and S.G. Terrace, "A Directory Retrieval System Based on Connected Letter Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 35.4).

[159] M. Cravero, L. Fissore, R. Pieraccini and C. Scagliola, "Syntax Driven Recognition of Connected Words by Markov Models", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 35.5).

[160] D. Jouvet and R. Schwartz, "One-Pass Syntax-Directed Connected-Word Recognition in a Time-Sharing Environment", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 35.8).

[161] J.-L. Gauvain and J. Mariani, "Evaluation of Time Compression for Connected Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 35.10).

[162] H. Ney, "The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition", *Proc. IEEE Trans. Acoust., Speech, and Signal Process., vol.ASSP*−32, pp.263-271, April 1984.

[163] J.L. Hieronymus and W.J. Majurski, "A Reference Speech Recognition Algorithm for Benchmarking and Speech Data Base Analysis", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1573-1576.

[164] S.C. Glinski, "On the Use of Vector Quantization for Connected-Digit Recognition", *AT&T Tech. J.*, vol.64, pp.1033-1045, May-June 1985.

[165] L.R. Rabiner and S.E. Levinson, "A Speaker-Independent, Syntax-Directed, Connected Word Recognition System Based on Hidden Markov Models and Level Building", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.561-573, June 1985.

[166] L.R. Rabiner, J.G. Wilpon and B.-H. Juang, "A Continuous Training Procedure for Connected Digit Recognition", *Proc. IEEE−IECEJ−ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1065-1068.

[167] K. Tajima, M. Komura and Y. Sato, "Connected Word Recognition by Overlap and Split of Reference Patterns and Its Performance Evaluation Tests", *Proc. IEEE−IECEJ−ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1101-1104.

[168] C. Scagliola and D. Sciarra, "Two Novel Algorithms for Variable Frame Analysis and Word Matching for Connected Word Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1105-1108.

[169] M. Watari, "New DP Matching Algorithms for Connected Word Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1113-1116.

[170] L.R. Rabiner, J.G. Wilpon and B.-H. Juang, "A Segmental k-Means Training Procedure for Connected Word Recognition", *AT&T Technical Journal*, vol.65, issue 3, pp.21-31, May/June 1986.

(See also [6,7,8,10,12,15,149,154,166,185,190,223].)

R.9: Speech recognition based on units smaller than the word

[171] R. Billi, "Vector Quantization and Markov Source Models Applied to Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.574-577.

[172] L.R. Bahl, A.G. Cole, F. Jelinek, R.L. Mercer, A. Nadas, D. Nahamoo and M.A. Picheny, "Recognition of Isolated-Word Sentences from a 5000-Word Office Correspondence Task", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, April 1983, pp.1061-1064.

[173] L.C. Sauter, "Isolated Word Recognition Using a Segmental Approach", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.850-853.

[174] D. Mergel and H. Ney, "Phonetically Guided Clustering for Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.854-857.

[175] F. Jelinek et al., "A Real-Time, Isolated-Word, Speech Recognition System for Dictation Transcription", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.858-861.

[176] M.S. Glassman, "Hierarchical DP for Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.886-889.

[177] H. Ney, "A Script-Guided Algorithm for the Automatic Segmentation of Continuous Speech", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1209-1212.

[178] H. Bourlard, Y. Kamp and C.J. Wellekens, "Speaker Dependent Connected Speech Recognition Via Phonemic Markov Models", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1213-

1216.

[179] J.-P. Brassard, "Integration of Segmenting and Nonsegmenting Approaches in Continuous Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1217-1220.

[180] A.M. Colla, C. Scagliola and D. Sciarra, "A Connected Speech Recognition System Using a Diphone-Based Language Model", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1229-1232.

[181] S. Haltsonen and P. Ruusunen, "Collection of Phoneme Samples Using Time Alignment and Spectral Stationarity of Speech Signals", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1561-1564.

[182] Y. Kobayashi and Y. Niimi, "Matching Algorithms Between a Phonetic Lattice and Two Types of Templates – Lattice and Graph", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1597-1600.

(See also [6,7,8,10,15,223,237,242,248,249,251].)

R.10: Specialised electronic hardware for speech recognition

[183] B. Ackland, N. Weste and D.J. Burr, "An Integrated Multiprocessing Array for Time Warp Pattern Matching", Eighth Annual Symposium on Computer Architecture, 1981 (published in *Sigarch Newsletter*, vol.9, no.3, pp.197-215).

[184] M.A. Yoder and L.J. Siegel, "Dynamic Time Warping Algorithms for SIMD Machines and VLSI Processor Arrays", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.1274-1277.

[185] J. Peckham, J. Green, J. Canning and P. Stephens, "Logos – A Real Time Hardware Continuous Speech Recognition System", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, May 1982, pp.863-866.

[186] H. Ishizuka, M. Watari, H. Sakoe, S. Chiba, T. Iwata, T. Matsuki and Y. Kawakami, "A Microprocessor for Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.503-506.

[187] J. MacAllister, "Systolic Arrays for Dynamic Programming in Speech Recognition Systems", *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, April 1983, pp.507-510.

[188] M.K. Brown, R. Thorkildsen, Y.H. Oh and S.S. Ali, "The DTWP: An LPC Based Dynamic Time Warping Processor for Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984

(paper 25B.5).

[189] R. Kavaler, R.W. Brodersen, T.G. Noll, M. Lowy and H. Murveit, "A Dynamic Time Warp IC for a One Thousand Word Recognition System", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 25B.6).

[190] J.G. Ackenhausen, "The CDTWP: A Programmable Processor for Connected Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 35.9).

[191] Y. Kitazume, E. Ohira and T. Endo, "LSI Implementation of a Pattern Matching Algorithm for Speech Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.1-4, February 1985.

[192] M.A. Yoder and L.H. Jamieson, "Simulation of a Highly Parallel System for Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1449-1452.

[193] T.S. Anantharaman and R. Bisiani, "Custom Data-Flow Machines for Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.1847-1850.

[194] J.R. Mann and F.M. Rhodes, "A Wafer Scale DTW Multiprocessor", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1557-1560.

[195] F. Charot, P. Frison and P. Quinton, "Systolic Architectures for Connected Speech Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-34, pp.765-779, August 1986.

R.11: Evaluation of commercial recognition systems

[196] G. Kaplan, R. Reddy and Y. Kato, "Words into action", *IEEE Spectrum*, vol.17, no.6, pp.22-29, June 1980.

[197] G.R. Doddington and T.B. Schalk, "Speech recognition: turning theory to practice", *IEEE Spectrum*, vol.18, no.9, pp.26-32, September 1981.

[198] W.A. Lea, "Selecting the Best Speech Recognizer for the Job", *Speech Technology*, vol.1, no.4, pp.10-29, January/February 1983.

[199] B. Rubinchek, "Towards Standards for Speech I/O Systems", *Speech Technology*, vol.1, no.4, pp.40-42, January/February 1983.

[200] D.W. Bell and R.W. Becker, "Designing Experiments to Evaluate Speech I/O Devices and Applications, *Speech Technology*, vol.1, no.4, pp.70-79, January/February 1983.

[201] J.M. Baker, D.S. Pallett and J.S. Bridle, "Speech Recognition Performance Assessments and Available Databases", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, April 1983, pp.527-530.

[202] J.P. Woodard and W.A. Lea, "New Measures of Performance for Speech Recognition Systems", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1984 (paper 9.6).

[203] P. Wallich, "Putting speech recognizers to work", *IEEE Spectrum*, vol.24, no.4, pp.55-57, April 1987.

[204] J.J. Mariani, "Speech Technology in Europe", *Proc. European Conf. Speech Technology*, September 1987, vol.1, pp.431-439.

[205] T.J. Thomas, "The Prediction of Speech Recogniser Performance by the Use of Laboratory Experiments: Some Preliminary Observations", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.245-248.

[206] J.K. Baker and J.M. Baker, "Large Vocabulary Natural Language Speech Recognition in Software", *Proc. European Conf. Speech Technology*, September 1987, vol.2, p.440.

[207] R.K. Moore, "Connected Digit Recognition in a Multilingual Environment", RSRE Memorandum No.4134, February 1988 (also published as Project II Final Report NATO AC/243 (Panel 3/RSG 10) D/11, 1987)

(See also [77,71,185,216].)

R.12: Applications of speech recognition

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

[209] J.B. Peckham, "Automatic speech recognition — a solution in search of a problem?", *Behaviour and Information Technology*, vol.3, no.2, pp.145-152, April-June 1984.

[210] A.F. Newell, "Communicating via speech — the able bodied and the disabled", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986, pp.1-7.

[211] J.A. Harrison, G.R. Hobbs, J.R. Howes, N. Cope and G.R. Wright, "Machine supported voice dialogue, used in training air traffic controllers", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986, pp.110-115.

[212] R.M. Stephens, "Voice recognition for the BBC microcomputer: an aid for physically handicapped children", *IEE Conf. Pub.* 258 (*Speech*

Input/Output; Techniques and Applications), March 1986, pp.230-233.

[213] A. Jackson and G. Waterworth, "An approach to speech technology in civil air traffic applications", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986, pp.237-241.

[214] M. Talbot, "Speech recognition: is it working?", *British Telecom Technology Journal*, vol.4, no.2, pp.62-68, April 1986.

[215] P.S. Kelway, "Putting Speech Technology to Work", conclusions drawn from ESPRIT Project 449 (1984-86).

[216] J.M. Baker, "State-of-the-Art Speech Recognition U.S. Research and Business Update", *Proc. European Conf. Speech Technology*, September 1987, vol.1, pp.440-447.

[217] C. Gagnoulet, F.Zurcher, J. Tirbois and T. Serradura, "PUBLIVOX: A Voice Controlled Card Pay Phone", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.61-64.

[218] C.R. Frankish and D.M. Jones, "Parcel Sorting by Speech Recognition: A Case Sudy in Vocabulary Design", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.197-201.

[219] M. Blomberg, K. Elenius, B. Lundström and L. Neovius, "Speech Recognizer for Voice Control of Mobile Telephone", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.210-213.

[220] J.M. Noyes and C.R. Frankish, "Voice Recognition – Where Are the End-Users?", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.349-352.

(See also [16,96,128,129,196,198,221,222,223,227,238,239].)

R.13: Human factors and user-system interface design

[221] J.A. Waterworth, "Man-machine speech 'dialogue acts'", *British Telecom Technology J.*, vol.1, no.1, pp.106-112, July 1983.

[222] J.A. Waterworth, "Interaction with machines by voice: a telecommunications perspective", *Behaviour and Information Technology*, vol.3, no.2, pp.163-177, April-June 1984.

[223] J.A. Waterworth, "Interaction with machines by voice – human factors issues", *British Telecom Technology J.*, vol.2, no.4, pp.56-63, September 1984.

[224] W.A. Ainsworth, "Audio Feedback for Error Correction in a Digit Recognition Task", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.65-68.

[225] R.G. Leiser, M. de Alberdi and D.J. Carr, "Generic Issues in Dialogue Design for Speech Input/Output", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.69-72 (and generic dialogue module specifications distributed at the conference).

[226] A.F. Starr, S.M. Hudson, D.M. Jones and C.R. Frankish, "Automatic Speech Recognition: The User's View", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.73-76.

[227] S.M. Furner, "Rapid prototyping as a design tool for dialogues employing voice recognition", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.353-356.

[228] K. Hapeshi, D.M. Jones and C. Frankish, "Human Factors Aspects of Template Training", *Proc. European Conf. Speech Technology*, September 1987, vol.2, pp.397-400B.

(See also [209,220,239].)

R.14: Hierarchical and progressive recognition decision procedures

[229] T. Kaneko and N.R. Dixon, "A Hierarchical Decision Approach to Large-Vocabulary Discrete Utterance Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-31, pp.1061-1066, October 1983.

[230] Z. Guo-tian, "On Associative Recognition of Isolated Chinese Word", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.37-40.

[231] K.-C. Pan, F.K. Soong, L.R. Rabiner and A.F. Bergh, "An Efficient Vector-Quantization Preprocessor for Speaker Independent Isolated Word Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, March 1985, pp.874-877.

[232] K.-C. Pan, F.K. Soong and L.R. Rabiner, "A Vector-Quantization-Based Preprocessor for Speaker-Independent Isolated Word Recognition", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-33, pp.546-560, June 1985.

[233] A.F. Bergh, F.K. Soong and L.R. Rabiner, "Incorporation of Temporal Structure Into a Vector-Quantization-Based Preprocessor for Speaker-Independent, Isolated-Word Recognition", *AT&T Tech. J.*, vol.64, pp.1047-1063, May-June 1985.

[234] R. Billi, G. Massia and F. Nesti, "Word Preselection for Large Vocabulary Speech Recognition", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.65-68.

[235] A. Aktas, B. Kammerer, W. Küpper and H. Lagger, "Large-Vocabulary Isolated Word Recognition with Fast Coarse Time Alignment", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.709-712.

[236] J. Miwa and K. Kido, "Speaker-Independent Word Recognition for Large Vocabulary Using Pre-Selection and Non-Linear Spectral Matching", *Proc. IEEE–IECEJ–ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2695-2698.

R.15: Adaptation of reference patterns

[237] S. Furui, "A Training Procedure for Isolated Word Recognition Systems", *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-28, pp.129-136, April 1980.

[238] D.L. Morrison and T.R.G. Green, "Adaptive Interface Techniques in Recognising Speech and Similar Inputs", Memo No.457, MRC/SSRC Social and Applied Psychology Unit, University of Sheffield, September 1981 (archived in *British Library Experimental Journal, Computer Human Factors*).

[239] T.R.G. Green, S.J. Payne, D.L. Morrison and A. Shaw, "Friendly interfacing to simple speech recognizers", *Behaviour and Information Technology*, vol.2, no.1, pp.23-38, January-March 1983.

[240] R.J. Golibersuch, "Automatic Prediction of Linear Frequency Warp for Speech Recognition", *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, April 1983, pp.769-772.

[241] R.I. Damper and S.L. MacDonald, "Template Adaptation in Speech Recognition", *Proc. Inst. of Acoust.*, vol.6, part 4, pp.293-299, 1984.

[242] J.S. Bridle and M.P. Ralls, "An Approach to Speech Recognition Using Synthesis-by-Rule", chapter 10 (pp.277-292) in F. Fallside and W.A. Woods (eds), *Computer Speech Processing*, Prentice/Hall International, 1985.

[243] Y. Niimi, N. Kitamura and Y. Kobayashi, "Speaker Adaptation in an Isolated Word Recognition System", *Studia Phonologica* XIX, 1985, pp.34-42.

[244] J.K. Goatcher and J.S. Mason, "An adaptive approach to a speaker-independent isolated word system with short training", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986,

pp.67-70.

[245] R.C. Power, R.D. Hughes and R.A. King, "Verification, archetype updating, and automatic token set selection, as a means of improving the performance of menu driven isolated word recognition systems using time encoded speech descriptors in high acoustic noise backgrounds", *IEE Conf. Pub.* 258 (*Speech Input/Output; Techniques and Applications*), March 1986, pp.144-151.

[246] J.M. Baker and D.F. Pinto, "Optimal and Suboptimal Training Strategies for Automatic Speech Recognition in Noise, and the Effects of Adaptation on Performance", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.745-748.

[247] B.P. Landell, R.E. Wohlford and L.G. Bahler, "Improved Speech Recognition in Noise", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.749-751.

[248] F. Togawa, M. Hakaridani, H. Iwahashi and T. Ueda, "Voice-Activated Word Processor with Automatic Learning for Dynamic Optimization of Syllable-Templates", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.1121-1124.

[249] M. Sugiyama, "Unsupervised Speaker Adaptation Methods for Vowel Templates", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2635-2638.

[250] K. Shikano, K.-F. Lee and R. Reddy, "Speaker Adaptation through Vector Quantization", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2643-2646.

[251] Y. Niimi and Y. Kobayashi, "Synthesis of Speaker-Adaptive Word Templates by Concatenation of the Monosyllabic sounds", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2651-2654.

[252] K. Choukri, G. Chollet and Y. Grenier, "Spectral transformations through Canonical Correlation Analysis for speaker adaptation in ASR", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2659-2662.

[253] K. Sugawara, M. Nishimura and A. Kuroda, "Speaker Adaptation for a Hidden Markov Model", *Proc. IEEE-IECEJ-ASJ Int. Conf. Acoust., Speech, and Signal Process.*, April 1986, pp.2667-2670.

[254] A.J. Hewett, G. Holmes and S.J. Young, "Dynamic Speaker Adaptation in Speaker-Independent Word Recognition", *Proc. Inst. of Acoust.*, vol.8, part 7, pp.275-282, 1986.

[255] F.R. McInnes, M.A. Jack and J. Laver, "An Isolated Word Recognition System with Progressive Adaptation of Templates", *Proc. Inst. of Acoust.*,

vol.8, part 7, pp.283-290, 1986.

[256] R. Meddis, "Towards an Auditory Primal Sketch", *Proc. Inst. of Acoust.*, vol.8, part 7, pp.589-596, 1986.

[257] K. Shikano, K.-F. Lee and R. Reddy, "Speaker Adaptation through Vector Quantization", Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, December 1986.

[258] F.R. McInnes, M.A. Jack and J. Laver, "Experiments with Template Adaptation in an Isolated Word Recognition System", *Proc. European Conf. on Speech Technology*, September 1987, vol.2, pp.484-487.

[259] F.R. McInnes and M.A. Jack, "Reference template adaptation in speaker-independent isolated word speech recognition", *Electronics Letters*, vol.23, no.24, pp.1304-1305, November 1987.

[260] F.R. McInnes, M.A. Jack and J. Laver, "Template adaptation in an isolated word recognition system", submitted for publication in *IEE Proc.—F (Commun., Radar and Signal Process.)*.

(See also [50,154,246].)

R.16: Statistical analysis of results

[261] M.R. Spiegel, *Theory and Problems of Probability and Statistics*, McGraw-Hill, 1975, p.161.

[262] L. Gillick, "Some Statistical Issues Involved in the Comparison of Speech Recognition Algorithms", handwritten notes distributed at six-monthly review meeting of the DARPA speech computing projects, BBN, Cambridge, Massachussetts, October 1987.

R.17: Other mathematical background

[263] T.M. Apostol, *Calculus*, Blaisdell Publishing Co., 1962, vol.II, pp.413-420.

INDEX OF AUTHORS

The numbers opposite each name refer to publications listed in the preceding reference list, of which the named person is an author or (in cases marked "(ed)") an editor.

PAPER 1

COMPARATIVE STUDY OF TIME SEGMENTATION AND SEGMENT REPRESENTATION TECHNIQUES IN A DTW-BASED WORD RECOGNISER

IEE Conference Publication No.258 (Proceedings of the IEE International Conference on "Speech Input/Output; Techniques and Applications", London, March 1986), pp.21-26

Note:- The statement made in this paper that Kuhn and Tomaschewski used unnormalised log filter energies in their trace segmentation procedure is incorrect: in fact, according to their paper (reference 4 here), and as stated in the main text of the thesis, they used filter energies which were normalised for overall energy in each frame but were not logarithmically transformed.

COMPARATIVE STUDY OF TIME SEGMENTATION AND SEGMENT REPRESENTATION TECHNIQUES IN A DTW-BASED WORD RECOGNISER

F.R. McInnes, M.A. Jack and J. Laver

Centre for Speech Technology Research, University of Edinburgh, UK

ABSTRACT

Results of experiments comparing segmentation techniques as preprocessing for a DTW-based isolated word recognition system are presented. Various features of these results, and those of previously reported experiments, are discussed. An application of segmentation techniques in an efficient multiple-pass recognition system is described.

INTRODUCTION

Many isolated and connected word recognition systems operate by comparison of the input speech with reference patterns (templates), where each template represents one word of the designated vocabulary. This comparison can be accomplished with an optimal non-linear time alignment by the dynamic programming technique known as dynamic time warping (DTW). (See for instance Itakura (1); Myers et al (2)).

In the basic DTW-based isolated word recogniser, the pattern representing each (reference or unknown input) word is a sequence of vectors of parameters representing short time sections (frames) of speech. The number of frames varies according to the duration of the word. Various researchers have applied preprocessing techniques to normalise all the words to a standard number of vectors before the DTW comparison. (This normalisation is helpful in allowing the computationally expensive search for the optimal alignment in the DTW stage to be considerably reduced (2), and is also necessary for some hardware implementations, such as that described by Brown et al (3)). In some cases this normalisation of the timescale is linear (2); in other cases a non-linear normalisation based on characteristics of the original pattern has been applied (Kuhn and Tomaschewski (4); Pieraccini and Billi (5); Chuang and Chan (6); Gauvain et al (7)). One form of non-linear word length normalisation which has been applied with some success is trace segmentation (4), in which the "trace" formed by connecting successive frame vectors by line segments in acoustic parameter vector space is divided into segments of equal length and a parameter vector is derived to represent each segment or each segment boundary. Three methods of deriving these vectors have been described. Two of these methods derive representations at segment boundaries - in one case by linear INTERPOLATION between the two frame vectors adjacent to the segment boundary (4); in the other case by SELECTION of whichever of those two vectors is closer to the boundary (7). The third method derives a vector to represent each segment, by AVERAGING the frame vectors within that segment (Ney(8)).

This present paper presents an experimental comparison of these different segment representation methods. The three methods - interpolation, selection and averaging - were applied both with linear time segmentation and with trace segmentation.

DESCRIPTION OF EXPERIMENTS

Speech data base

Two English-language vocabularies were selected for the experiments: one comprising the digits from 0 to 9 (with 0 pronounced "zero"), and one consisting of 20 mostly disyllabic or polysyllabic words (against, begin, evergreen, flowering, following, framework, horizontal, Japanese, possible, remaining, retaining, single, sometimes, spring, susceptible, these, those, trained, training, year).

Recognition tests were carried out separately, in a speaker-dependent mode, for each of three speakers (two male and one female). Each speaker first read each vocabulary aloud several times (five times for the digits, and three times for the other vocabulary), in a fixed order, to provide templates for the recogniser, and then read out a further five repetitions, in which the order of the words was varied from one repetition to another, to provide test data. All the utterances were recorded in a quiet environment using a fixed microphone.

The recordings were lowpass filtered at 5 kHz and digitised at 10 kHz. The beginning and end of each word were located by visual inspection of a display of the waveform.

Isolated word recogniser

For each reference or test word, an acoustic analysis is performed to derive vectors of eight mel frequency cepstrum coefficients (as defined by Davis and Mermelstein (9)) at intervals of 12.8 ms. The resulting sequences of vectors are used directly as input for the DTW comparison, or else processed to obtain segment representations. If trace segmentation is applied, the distances along the trace can be measured using either the Euclidean norm or the absolute value norm. In fact, the absolute value norm was used in the experiments reported here, since it was found in preliminary tests to yield better results. The DTW routine incorporates Itakura's local path constraints (1,2) with type (c) weighting (2) and with the test word along the x-axis. The absolute value norm is used as the vector distance measure. In these experiments no global path constraint (warping window or band) was imposed, but endpoint constraints were strictly observed.

Recognition experiments

For each vocabulary and speaker, various preprocessing conditions were defined. For each set of preprocessing conditions, recognition tests were carried out on the five test repetitions of the words using each of the training repetitions in turn to provide the templates. The results were averaged over the different choices of template set.

A set of preprocessing conditions consisted of normalisation to N segments per word, for one of various values of N, by one of six combinations of segmentation and segment representation techniques. The six combinations resulted from the choice of linear time segmentation or trace segmentation and the three possible methods of segment (or segment boundary) representation. In each case, the same preprocessing was applied both to the templates and to the test utterances.

For each vocabulary and speaker, recognition tests were also carried out (again using each of the template sets in turn) with no segmentation of the test and reference words.

RESULTS

The results of the experiments with word length normalisation are plotted, for respective vocabularies and segmentation procedures, in figures 1-4. Each of these diagrams shows word recognition error rate (averaged, for each set of preprocessing conditions, over all speakers and over all the sets of templates for each speaker) plotted against the average number of vectors used to represent each word. (The experiments with the digits, and some with the other vocabulary, showed that the selection method was inferior to the interpolation and averaging methods, and therefore full experiments with the other vocabulary were conducted using only interpolation and averaging). Also plotted is a measure R of the quality of discrimination between correct and incorrect words' templates. This is defined by

$$R = \frac{(\text{mean value of } r) - 1}{\text{standard deviation of values of } r}$$

where, for one recognition of one test word, r is the ratio of the smallest word distance obtained for an incorrect-word template to the distance obtained for the correct word's template. (The word is recognised correctly if r>1). The value of R was computed separately for each template set, and then averaged in the same way as the error rate.

The results obtained without preprocessing are plotted at the right of each figure, and are marked by horizontal broken lines to allow comparison with those obtained with preprocessing.

It will be seen from the graphs that, in the case of the averaging method of segment representation, the average number of vectors per word is not always an integer. This is because, as part of the segmentation and averaging procedure, when none of the original frame vectors falls within a particular segment, that segment is extended to include the next frame vector (and the start of the next segment is moved along correspondingly): thus the final number of segment representations obtained for a word may be less than the initially specified value of N. This occurs particularly for values of N which approach (or exceed) the number of frames per word. It also occurs more with trace segmentation than with linear time segmentation. Where either of the other two segment representation techniques is applied, the number of vectors per word is N+1, because there are N+1 segment boundaries (including the initial and final ones).

DISCUSSION

General comments on the results

These results have been obtained from a fairly small data base, consisting of a total of 780 isolated word utterances by only three speakers. Continuation of this work to include more data from different speakers could improve the reliability of the conclusions drawn in the paragraphs below, some of which can only be tentative with the existing results.

The results varied considerably from speaker to speaker, especially for the digits, where, for example, the three speakers' error rates with no preprocessing were 6.8%, 24.8% and 11.6% (average 14.4%). (The corresponding figures for the other vocabulary were 14.7%, 12.3% and 11.3% respectively (average 12.8%). In each case the first two are the male speakers' results). The average word length before segmentation also varied among the speakers for each vocabulary, by as much as 20% in the case of the 20-word vocabulary.

Overall the recognition accuracy attained was rather poor. This was partly because each template was derived from only one training utterance; further experiments with the same data showed that the error rates could be nearly halved by deriving each template from two or three of the training utterances using an averaging procedure.

It should be noted in interpreting the results that the discrimination measure R is less subject to statistical variation than the percentage error rate; but also that the expected value of R is non-linearly related to the expected error rate, and so the averaging of R over different speakers or different sets of templates, where the quality of recognition varies considerably, will not necessarily give an accurate indication of the expected average recognition performance.

Comparison of segmentation techniques

The results for trace segmentation (figures 2 and 4) are similar on the whole to those obtained using linear time segmentation (figures 1 and 3). This is surprising in view of the results of previous experiments (6,7) which indicated that trace segmentation gave substantially better recognition performance. It is, however, in agreement with Ney's connected word recognition results (8). A possible explanation for this discrepancy lies in the fact that both Ney's experiments and those reported in the present paper were obtained using cepstral coefficients, whereas some of the other

trace segmentation experimenters (4,7) used log bandpass filter energies. (Chuang and Chan (6) used LPC). One difference between these two types of acoustic representation is that log filter energies (used in unnormalised form for trace segmentation (4)) contain some information as to the overall energy in each frame of speech, whereas cepstral coefficients do not (since the zeroth order coefficient is omitted). To investigate this hypothesised explanation, some experiments were conducted on part of the data base, in which the segment boundaries were determined by trace segmentation using log spectral coefficients (corresponding to 32 bandpass filters), and representations for the segments thus defined were derived as before from the cepstral coefficients. The recognition results (for the 20-word vocabulary spoken by one male speaker) were consistently poorer than those obtained using trace segmentation based on the cepstral coefficients. However, this may indicate merely that the use of different acoustic representations for the trace segmentation and word comparison phases introduces recognition errors.

The superiority of linear time segmentation or trace segmentation may depend partly on the nature of the vocabulary to be recognised: on the digits, which are mostly monosyllabic words (in which less non-linear timescale variation between utterances may be expected), the best results were obtained using linear time segmentation, whereas on the other vocabulary, which includes many words of two, three or four syllables, trace segmentation gave better results.

For each vocabulary, both linear time segmentation and trace segmentation (with interpolation) yielded an improvement in recognition over what was obtained with no preprocessing, where the average numbers of vectors per word were similar. (On average, over the two vocabularies and the two techniques, the reduction in error rate was about 0.6%, and the increase in the value of R was about 0.07). This suggests that there is some advantage in normalising all words to the same number of vectors - besides the benefits mentioned in the introduction above. This agrees with the results of (6) and (7) on fixed and variable length segmentation, though not with those of (5).

Comparison of segment representation techniques

The results for digit recognition plotted in figures 1 and 2 show that selection of one of the original frame vectors at each segment boundary leads to recognition performance consistently worse than is obtained when vectors are interpolated. (The same phenomenon was observed when these techniques were compared for the other vocabulary, though in this case the results for the selection technique are not plotted here because it was applied to the words of only one of the speakers). This is as might be expected, since the selection of the nearest original vector gives only a rough approximation to a representation for the segment boundary. Even with selection, however, the best results were slightly better than those obtained with no preprocessing of the original vector sequences.

The comparison of averaging with the other two techniques is less straightforward. When the number of segments per word (N) is small (less than about half the average number of frames per word), averaging is clearly better than interpolation (or selection). This is what could be expected, since, when a segment contains more than two of the original vectors, averaging is the only one of the three techniques to make use of the information contained in all the vectors. When N becomes larger, however, interpolation produces better results than averaging. This may be attributed to the fact that interpolation takes account of the exact positions of the segment boundaries, rather than just of which frame vectors are contained in each segment. But also the present formulation of the averaging procedure, which adjusts the segment boundaries when a segment contains no frame vector, is probably not optimal - particularly as it leads to variations in the number of vectors per word after segmentation. A better procedure might be to interpolate a vector at the centre of any segment which contains no vectors.

It is worth noting that, using the averaging method, performance fairly similar to that with no segmentation was obtained when the number of vectors per word was reduced to about a third of its original value. This suggests that the frame rate of the original analysis (78.125 frames per second) was probably unnecessarily high.

Application of segmentation in a multiple-pass recogniser

Computational efficiency is a major consideration in applications of DTW-based word recognition. The basic DTW recogniser compares each input word with all the templates in turn, which is computationally costly especially for large vocabularies. Among the modifications proposed to reduce the computational load is the application of a preliminary simple comparison to eliminate poorly-matching templates before the DTW stage. (See for instance Kaneko and Dixon (10); Pan et al (11)). This can be achieved very easily in a recognition system with a segmentation capability: the initial comparison can be made using averaged representations for a small number of segments per word (2 seems a good choice for the number of segments here, from the above results); then only those templates giving distances within a prespecified factor of the smallest word distance need be considered in the more computationally intensive comparison with a larger number of segments per word.

This two-pass recognition procedure can be extended to a multiple-pass procedure using progressively more refined comparisons to eliminate more of the templates. The overall accuracy obtained may be better than in a conventional (less efficient) single-pass DTW recogniser without segmentation, on account of the improvements noted above as resulting from the preprocessing. It is also possible that an increase in accuracy can be obtained by making the final recognition decision (in

cases of doubt) by a procedure combining different segmentation conditions. Moreover, the number of vectors per word and the elimination threshold at each pass can easily be adjusted, to give the desired tradeoff between accuracy and speed for any particular vocabulary and application.

Experiments are currently in progress with a flexible multiple-pass recognition system using segmentation and DTW, to determine the effects of different numbers of passes and various values of the parameters at each pass.

ACKNOWLEDGEMENT

The work reported in this paper was enabled by support from the Science and Engineering Research Council.

REFERENCES

1. Itakura, F., 1975, "Minimum Prediction Residual Principle Applied to Speech Recognition", IEEE Trans. Acoust., Speech, and Signal Process., ASSP-23, 67-72.

2. Myers, C.S., Rabiner, L.R., and Rosenberg, A.E., 1980, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", IEEE Trans. Acoust., Speech, and Signal Process., ASSP-28, 623-635.

3. Brown, M.K., Thorkildsen, R., Oh, Y.H., and Ali, S.S., 1984, "The DTWP: An LPC Based Dynamic Time Warping Processor for Isolated Word Recognition", Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process., March 1984, paper 25B.5.

4. Kuhn, M.H., and Tomaschewski, H.H., 1983, "Improvements in Isolated Word Recognition", IEEE Trans. Acoust., Speech, and Signal Process., ASSP-31, 157-167.

5. Pieraccini, R., and Billi, R., 1983, "Experimental Comparison Among Data Compression Techniques in Isolated Word Recognition", Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., April 1983, 1025-1028.

6. Chuang, C.-K., and Chan, S.W., 1983, "Speech Recognition Using Variable Frame Rate Coding", Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., April 1983, 1033-1036.

7. Gauvain, J.L., Mariani, J., and Lienard, J.S., 1983, "On the Use of Time Compression for Word-Based Recognition", Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., April 1983, 1029-1032.

8. Ney, H., 1983, "Experiments in Connected Word Recognition", Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., April 1983, 288-291.

9. Davis, S.B., and Mermelstein, P., 1980, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences", IEEE Trans. Acoust., Speech, and Signal Process., ASSP-28, 357-366.

10. Kaneko, T., and Dixon, N.R., 1983, "A Hierarchical Decision Approach to Large-Vocabulary Discrete Utterance Recognition", IEEE Trans. Acoust., Speech, and Signal Process., ASSP-31, 1061-1066.

11. Pan, K.-C., Soong, F.K., and Rabiner, L.R., 1985, "A Vector-Quantization-Based Preprocessor for Speaker-Independent Isolated Word Recognition", IEEE Trans. Acoust., Speech, and Signal Process., ASSP-33, 546-560.

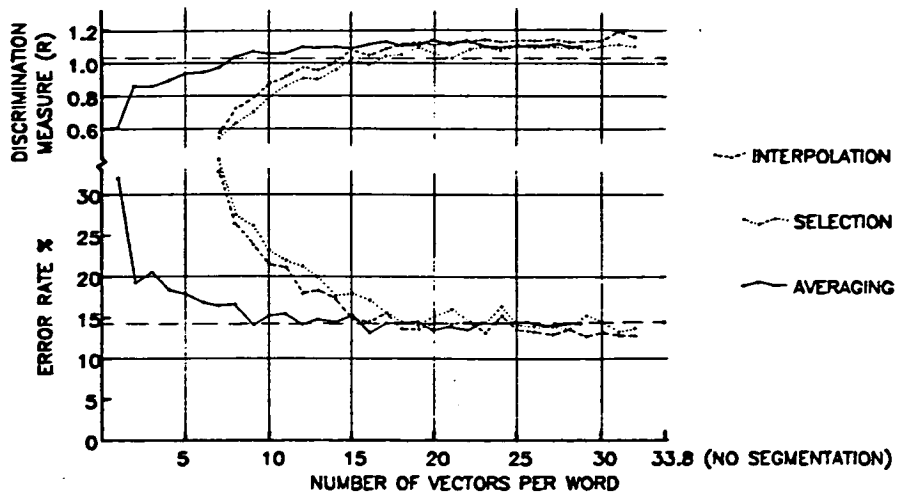


Figure 1 Results for recognition of digits using linear time segmentation

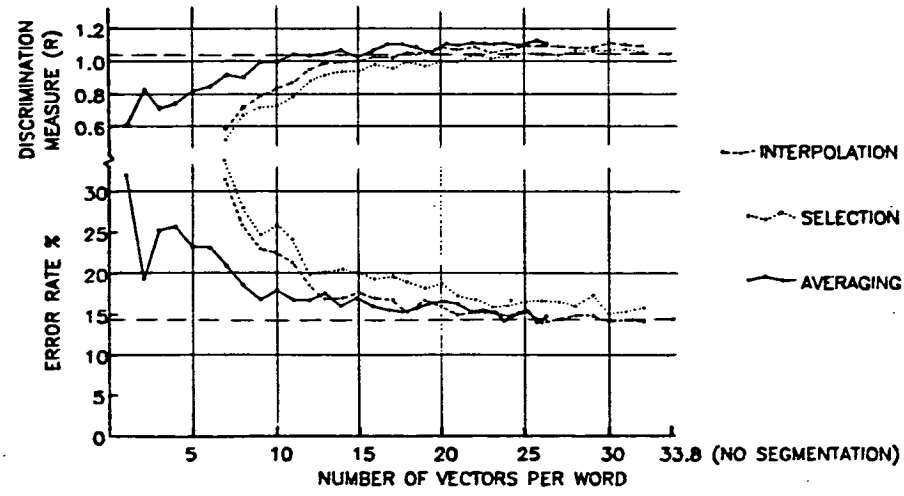


Figure 2 Results for recognition of digits using trace segmentation

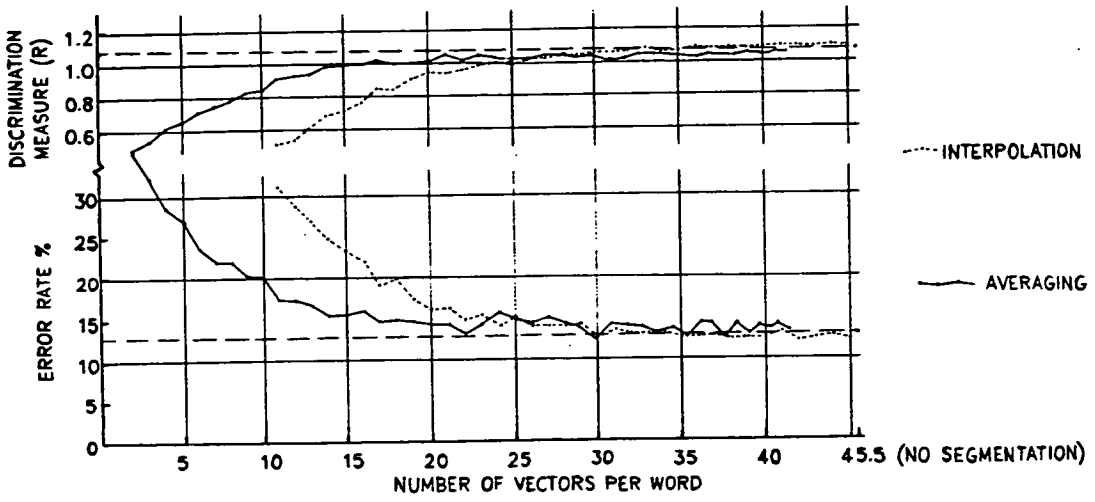


Figure 3 Results for recognition of the 20-word vocabulary using linear time segmentation

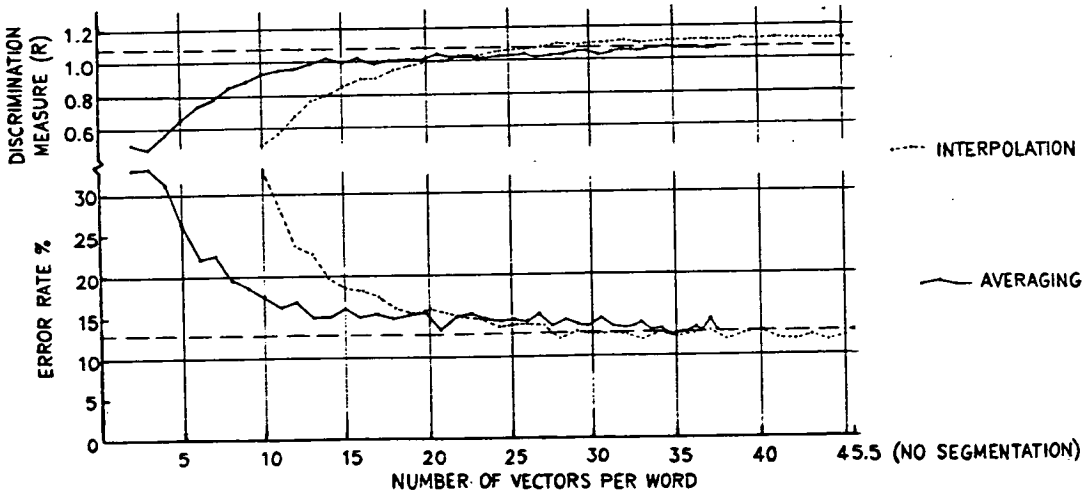


Figure 4 Results for recognition of the 20-word vocabulary using trace segmentation

PAPER 2

AN ISOLATED WORD RECOGNITION SYSTEM WITH PROGRESSIVE ADAPTATION OF TEMPLATES

Proceedings of the Institute of Acoustics, vol.8, part 7
(Proceedings of the Institute of Acoustics Autumn Conference,
Windermere, November 1986), pp.283-290

AN ISOLATED WORD RECOGNITION SYSTEM WITH PROGRESSIVE ADAPTATION OF TEMPLATES

F.R. McInnes, M.A. Jack and J. Laver

Centre for Speech Technology Research, University of Edinburgh

INTRODUCTION

The use of whole-word templates (reference patterns), obtained by a training procedure from utterances of the words to be recognised, is a well-established and successful approach to automatic recognition of isolated words from a small to medium-sized vocabulary. Each unknown input word is compared with all the stored templates, and is recognised as the word whose template yields the smallest value of a "word distance" or dissimilarity measure. Each template, and each input word, is represented for this comparison by a sequence of vectors of acoustic parameters (such as bandpass filter energies or cepstral coefficients), each vector being derived from a short time segment of the speech signal [1,2].

There are various problems which arise with this word recognition procedure, because of the degree of variability that occurs among utterances of the same word, by the same speaker on different occasions or (even more) by different speakers.

Temporal variation, and DTW matching

One form of variability is in the timescale of a word. The overall duration of the word varies from one utterance to another; also the relative durations of its parts (e.g. phones or syllables) vary. To cope with this temporal variation, the comparison procedure employs the dynamic programming technique known as dynamic time warping (DTW) [1,3], which finds the optimal alignment of a given pair of input and reference patterns, together with the corresponding word distance.

The main drawback of DTW is that it is computationally expensive; the amount of computation required is directly proportional to the number of templates to be matched, and to the square of the number of vectors per word. Various modifications have been proposed to reduce the computational requirements. Among these is the application of a relatively simple preliminary comparison to eliminate templates which are very dissimilar to the input word, so that only the most likely candidates are subjected to full DTW matching [4]. It is also possible to reduce the computation for each DTW matching operation by first compressing the representation of each word to a small number of acoustic vectors: various segmentation techniques exist which can be used to accomplish this [5,6].

These ideas of segmentation to compress word representations and of elimination of unlikely words by a simple comparison can be combined, as described below, to build a multiple-stage recognition system which achieves a substantial reduction of the time required to recognise each word, with little or no loss of accuracy, relative to the basic single-stage DTW-based recogniser.

Other forms of variability, and template adaptation

The effectiveness of a template-based word recogniser depends on its having good templates for all the words in the designated vocabulary. If the vocabulary is small, and the speaker and conditions are consistent, this can be achieved by deriving a template for each word from several utterances provided by the prospective user of the system during an initial training (enrolment) session [7]. However, if

the vocabulary is large, or there are frequent changes of speaker, this requirement of training becomes burdensome and time-consuming. In these cases, an alternative is to use a speaker-independent set of templates, generated by a representative set of speakers [8]. A speaker-independent system, to achieve satisfactory performance, requires several templates per word [8]; this, however, increases the computational requirements for the template matching process. A further disadvantage of a speaker-independent recogniser is that, when a new word is added to the vocabulary, it must be spoken by a representative set of speakers to train the system, in order to maintain the desired standard of recognition accuracy.

A method of improving the performance of a suboptimally trained word recognition system, whether speaker-trained or speaker-independent, is to incorporate adaptation of the templates during the recognition session [9]. The user can start using the recogniser with a small set of speaker-independent templates, or a set of single-utterance templates generated in a short training session, and the system will improve the templates by adapting them to the recognised input, so that its accuracy increases as it is used. This adaptation will also keep track of gradual changes in the speaker's voice or the background noise or transmission conditions.

The adaptation can be supervised (conditional on feedback as to the correctness of the recognition) or unsupervised. It may be helpful, especially in the case of unsupervised adaptation, to impose some condition as to the closeness of the word match or the certainty of the recognition decision before allowing a word to be used in adaptation of the best-matching template. A further option in the case of supervised adaptation is to implement negative adaptation in instances of incorrect recognition, so that the template becomes less similar to the input word which has been misrecognised, thus making the recurrence of the same error less likely.

Various techniques for template adaptation have been proposed [9,10]. The technique considered in this paper is a fairly straightforward one, in which a weighted averaging process is applied to the existing template and the input word. This adaptation technique has been incorporated into the multiple-stage recogniser already mentioned. The remaining sections of this paper contain a description of the system, the results of some preliminary experiments into possible adaptation options and an indication of directions for intended further research.

WORD RECOGNITION SYSTEM

The overall structure of the recognition system is shown in figure 1. The subsections below describe the components of the system and the operation of the multiple-stage decision and adaptation procedures.

Data acquisition, acoustic analysis and endpoint detection

The system is implemented in software on a Masscomp MC5500 minicomputer, using a built-in analogue-to-digital convertor for data acquisition, and an AP501 array processor to perform acoustic analysis. During training, interactive recognition or test data collection, the speaker is prompted, by visual and audible signals from a terminal, to utter each word during an interval of 1.5s. The speech is low-pass filtered at 8kHz, and digitised at a 20kHz sampling rate. The beginning and end of the word are located automatically using thresholds on energy and zero-crossings in 10ms frames. (If the number of words detected in the 1.5s interval is not exactly 1, the speaker is prompted to repeat the word.) After this endpoint detection, the speech signal is subjected to preemphasis (factor 0.98), and to 8th-order LPC analysis in a 25.6ms Hamming-windowed frame every 10ms, and 8 cepstral coefficients are derived to represent each frame.

AN ISOLATED WORD RECOGNITION SYSTEM

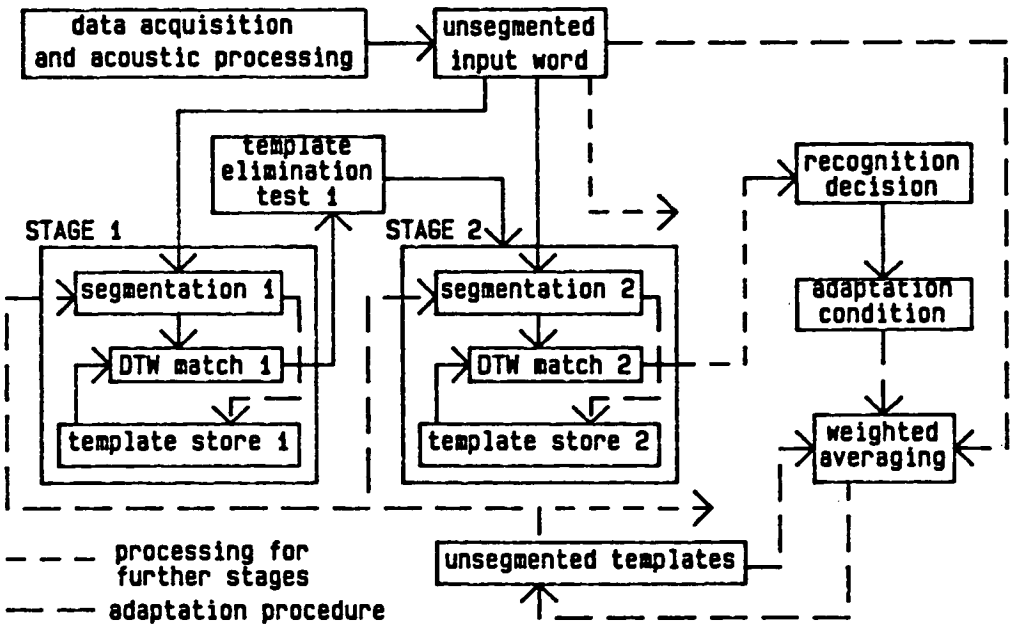
Word comparison technique

In each stage of the recognition process, the input word is segmented and compared with the (similarly segmented) templates for the words of the vocabulary under consideration. (At the first stage, all the templates are used; at later stages, some of them may have been eliminated.)

The segmentation technique involves dividing each word into a fixed number, N, of segments, and either averaging the acoustic vectors in each segment (so that the pattern after segmentation consists of N vectors) or interpolating a vector at each segment boundary (which generates N+1 vectors, including those at the beginning and end of the word). Either linear time segmentation or a form of trace segmentation [5] can be used. (Previous experiments comparing these segmentation techniques have been reported elsewhere [11].)

The segmented input word is compared with each template by DTW using Itakura's form of local path constraints and type (c) weighting [3], with the input word along the x-axis. The vector distance function in the DTW matching is the absolute value distance. (A pseudotemplate frame [12], with a constant distance to any input vector, which can be matched to any number of successive input vectors, is optionally appended before and after each template, to adjust for the possible inclusion by the endpoint detector of intervals before and after the input word.) This results in a word distance for each template matched.

FIGURE 1: STRUCTURE OF RECOGNITION SYSTEM



Multiple-stage decision procedure

The system incorporates a number of word comparison stages with different segmentation parameters. The number of stages, the details of each stage and the condition after each non-final stage for passing on templates for further comparison can easily be adjusted each time the recogniser is used. For the experiments reported here, the number of stages was fixed at 3, with segmentations resulting in 2, 10 and 30 vectors per word; the pseudotemplate frame technique was included in

the DTW at the third stage.

Appropriately segmented versions of all the templates are derived at the beginning of the recognition session and stored for use in the comparison stages. When the input word has been processed by the acoustic analysis into a sequence of vectors, this (unsegmented) word pattern is stored temporarily. In the first stage, the first segmentation is applied to the input word and it is matched by the DTW algorithm to the first segmented version of each template. The output of this stage is a set of word distances, one for each template. Let the distance obtained for template v be D_v; and let v^* be the value of the template index v that minimises D_v. Then the condition for passing template v on to the next stage is that

$$D_v < t_1 D_{v^*}, \qquad (1)$$

where t_1 (>1) is the threshold for the first stage. If only one value of v (i.e. v^*) satisfies (1), the input word is recognised as the word represented by template v^*. In this case, the remaining stages are not required for this word. Otherwise, the second segmentation is applied to the input word, and it is compared with the second segmented version of each template whose index v satisfies (1).

The output of the second stage is, like that of the first, a set of word distances. If there is no third stage in use, the input word is now recognised as the word whose template yields the smallest word distance in the second-stage comparison. If there is a third stage, a template retention criterion similar to (1), with a different threshold t_2, is applied to the second-stage word distances, and the templates satisfying this condition are passed on to the third stage. As before, if only one template satisfies the condition, the recognition decision is made and no further input segmentation or comparison is required.

Subsequent stages, if these exist, are similar to the second stage: at each stage, the appropriate segmentation is applied to the input word, and it is compared with the similarly segmented versions of those templates not eliminated by preceding stages. At some stage a recognition decision is reached.

It is possible to include a "rejection" or "no recognition" option, in which no recognition decision is made for the current input word if at any stage the ratio of the second-best to the best word distance is less than a set threshold. The rejection threshold can take a different value at each stage.

Template adaptation

The recognition procedure, when the recogniser is being used in its primary, interactive mode, is as follows. Once an input word has been recognised, the recognised word is printed out on the terminal screen. If the verification option is in use, the user is prompted for an indication of the correctness or incorrectness of the recognition. If it is incorrect, the second-best candidate word is displayed, and again the user is asked to verify its correctness. When a recognition is acknowledged as correct, or when both the best and the second-best candidates have been dismissed as incorrect, the system prompts for the next input utterance.

There is also a simulation option (used for the experiments described below), in which verification is achieved using a table of input word identities.

Template adaptation is applied whenever a recognition decision is reached and certain conditions are satisfied. Conditions which may be imposed are the following:-

Correctness of recognition: as confirmed by the user's response, or by reference to the input word identity file.

Word distance ratio: the ratio of the second-best to the best word distance, at the stage at which the decision is reached, must exceed a threshold. (This

threshold is not specified separately for each stage of comparison; but higher thresholds at the earlier stages are in effect imposed by specifying sufficiently high thresholds for template elimination in the recognition procedure.)

The main purpose of the distance ratio condition is to prevent adaptation in cases where there is no verification of the recognition and the degree of certainty of its correctness is low.

If there is verification (i.e. the adaptation is supervised), so that the correctness condition can be imposed. then there is also an option of negative adaptation, to make the template less like the misrecognised word; and not only the best candidate template, but also (where the best candidate is incorrect) the second-best, can be adapted, positively or negatively depending on whether it is correct.

The adaptation procedure consists of DTW alignment of the unsegmented versions of the template to be adapted and of the input word, and weighted averaging of each pair of vectors thus matched together, and interpolation of the vectors of the adapted template at integer points on a weighted-average timescale, as described in [7]. The weight on the input word is a constant, W, in the range from 0 to 1; the weight on the existing template is 1-W. ($W=0$ corresponds to no adaptation; $W=1$, to replacement of the template by the input word.) In negative adaptation, the procedure is the same, but W is negative (and so 1-W is greater than 1). When a template has been adapted, segmented versions of it are derived, replacing the previous versions, for use at all the stages of the recognition procedure.

EXPERIMENTS AND RESULTS

The adaptive recognition experiments reported here involve speaker-dependent recognition of utterances of the 10 English digits. Results have been obtained, to date, for two male speakers.

Speech data

Each set of templates, consisting of one for each digit, was formed in an interactive training session by a robust averaging procedure with DTW alignment. (The average number of utterances required per word of the vocabulary, to obtain two sufficiently similar ones, was about 3.) In these experiments two sets of templates for speaker 1 (designated R1A and R1B) and one set for speaker 2 (R2) were used.

The test data for adaptive recognition consisted of digit utterances collected in sets of 50 on separate occasions using the automatic data collection procedure mentioned above. The same sequence, containing 5 repetitions of each digit, was displayed and pronounced in each of these data collection sessions. For speaker 1, there were 10 data collection sessions over a period of nearly three weeks, providing 500 test utterances (designated T1). For speaker 2, 300 utterances (T2) were obtained in 6 sessions on successive working days. In each case, the templates were formed during the first few days of the data collection period.

All utterances, both for template formation and for testing, were recorded using a Sennheiser HME1019 headset microphone in a computer terminal room. There was a low level of continuous background noise, and there were also people working at nearby terminals during some of the sessions.

Adaptation parameters and results

The words in data set T1 were recognised using each of template sets R1A and R1B, and those in T2 were recognised using R2. The templates were adapted during the recognition process. Figures 2 and 3 show, for various adaptation

parameter values, the average word recognition accuracies obtained (over 1300 recognitions in all: 500, 500 and 300 with the respective template sets). Figure 2 shows results for adaptation with verification, with no distance ratio condition, with a number of combinations of positive and negative adaptation weight values. Figure 3 shows the performance using adaptation without verification, with and without a distance ratio condition. In the cases with verification, the second-best candidate template was also adapted when the first candidate was incorrect. The performance with no adaptation is shown as the first point marked "0" in each figure.

FIGURE 2:
RESULTS WITH SUPERVISED ADAPTATION

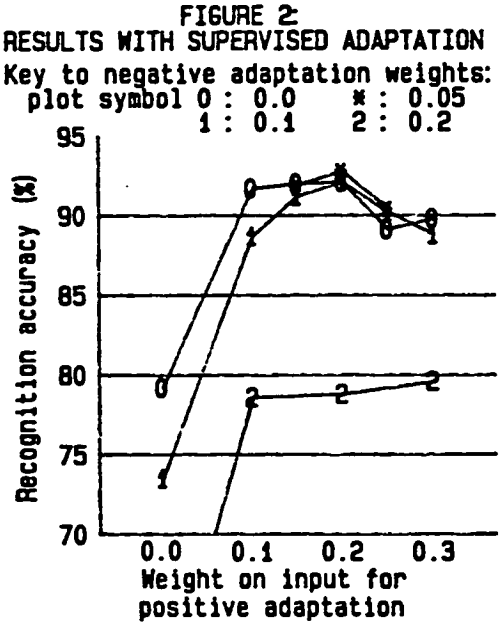
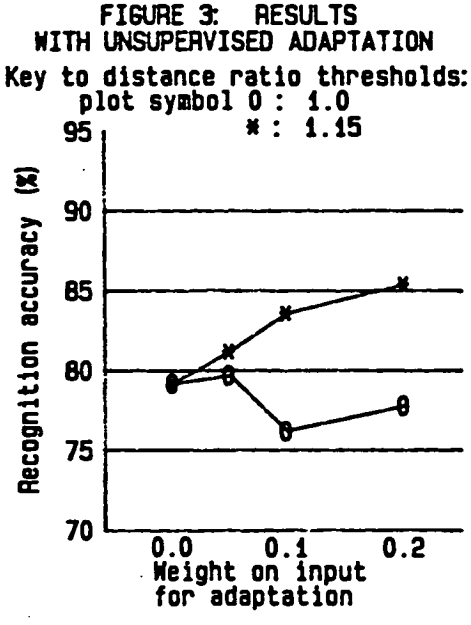


FIGURE 3: RESULTS WITH UNSUPERVISED ADAPTATION



The best recognition performance was obtained using supervised adaptation, with weights of 0.2 and -0.05 on the input in positive and negative adaptation. The best result for unsupervised adaptation was obtained when the adaptation was conditional on a word distance ratio exceeding 1.15 and the input weight was 0.2. (These values may not be optimal, as only two distance ratio thresholds and three weights have been tested for unsupervised adaptation.) The rates of correct recognition were 79.2% without adaptation (69.2%, 91.0% and 76.3% for the individual combinations of test data and templates); 92.8% (94.2%, 95.0%, 87.0%) with the optimal supervised adaptation; and 85.4% (85.2%, 92.4%, 74.3%) with the best unsupervised adaptation. The improvement is greater for R1A than for R2A: before adaptation the performance of R1A was considerably poorer, but with adaptation the two template sets yielded similar results. The poor results for speaker 2, even with adaptation, suggest that many of the errors for this speaker were due to deficiencies in the test data rather than in the templates.

DISCUSSION

The results obtained thus far indicate the usefulness of the template adaptation technique in enhancing speaker-dependent isolated word recognition performance. In particular, the adaptation procedure with verification can improve the

performance of a poor set of templates (such as R1A) to an apparently near-optimal level. More detailed examination of the results indicates that most of the improvement in the templates has occurred after about 50 input words. (This suggests that about 5 utterances of each word of the vocabulary are required for effective adaptation; but further experiments will be necessary to establish a more accurate estimate.)

The negative adaptation for misrecognised input appears to be of some benefit, but only if the negative weight is kept small and it is used in conjunction with the positive adaptation. It might be helpful to impose some word distance condition on negative adaptation, or to apply it only where the second-best candidate was correct: this could prevent adaptation away from noisy or badly detecteed input.

Even without feedback for verification, template adaptation can still improve the system's performance - though in this case it is preferable to have a threshold imposed on the ratio of the best two word distances, to prevent adaptation in cases of uncertainty. The choice of the distance ratio threshold is significant: if it is set too low, there is a risk that a template will be adapted repeatedly to utterances of the wrong word, resulting in severely degraded recognition performance on the misrecognised word and on the word that the template is intended to represent. More extensive experiments will be required to show whether it is possible to prevent this instability from arising over long sequences of input words. (If this cannot be guaranteed, a retraining procedure will have to be provided: see below.)

Further research is planned to extend the above results to more repetitions of the same words; to other vocabularies; to a larger number of speakers; to isolated word recognition using speaker-independent initial templates; and to connected word recognition, with initial templates derived from isolated utterances or a limited set of embedded utterances. There are also various options using multiple templates which could be explored: for instance, in a connected word recognition system, the adaptation procedure might be employed to generate from each word's initial template a set of adapted templates corresponding to contextual variations.

In applications of speech recognition, an important field of investigation is the interaction between the user and the system. The interactive recognition mode of the system that has been developed will allow experiments to be carried out in which the user can adapt to the recogniser as well as vice versa. The interactive mode allows more flexibility in the operation of the system, as the user can repeat words which are wrongly recognised, and, in the event of repeated failure to recognise a particular word, can abandon an existing template and generate a new one by providing one or more fresh training utterances of the word. (In practice, a user of a word recognition system is unlikely to tolerate very poor performance on particular words of the vocabulary - especially if each word has to be repeated until it is recognised correctly. So it is preferable to have a retraining procedure available for use as required during the recognition session.) The assessment of a system's performance becomes more complex as the degree of interaction between system and user increases; but this interaction is such an important feature of any application of a speech recogniser as to merit investigation despite this difficulty.

SUMMARY

An implementation of an isolated word recognition system incorporating a multiple-stage decision procedure and template adaptation has been described. Preliminary results of experiments with this system indicate that the template adaptation procedure can greatly improve recognition accuracy, especially where the initial set of templates gives poor performance. There is scope for further

investigation of a number of aspects of the adaptive recognition process, and for application of the adaptation technique to speaker-independent and multiple-template systems and to connected word recognition.

ACKNOWLEDGEMENT

The work reported in this paper was made possible by support from the Science and Engineering Research Council.

REFERENCES

[1] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-23, 67-72 (1975).

[2] G.M. White and R.B. Neely, "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering, and Dynamic Programming", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-24, 183-188 (1976).

[3] C.S. Myers, L.R. Rabiner and A.E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-28, 623-635 (1980).

[4] T. Kaneko and N.R. Dixon, "A Hierarchical Decision Approach to Large-Vocabulary Discrete Utterance Recognition", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-31, 1061-1066 (1983).

[5] M.H. Kuhn and H.T. Tomaschewski, "Improvements in Isolated Word Recognition", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-31, 157-167 (1983).

[6] R. Pieraccini and R. Billi, "Experimental Comparison Among Data Compression Techniques in Isolated Word Recognition", Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., April 1983, 1025-1028.

[7] R. Zelinski and F. Class, "A Learning Procedure for Speaker-Dependent Word Recognition Systems Based on Sequential Processing of Input Tokens", Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., April 1983, 1053-1056.

[8] L.R. Rabiner, S.E. Levinson, A.E. Rosenberg and J.G. Wilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques", IEEE Trans. Acoust., Speech, and Signal Process., vol. ASSP-27, 336-349 (1979).

[9] R.I Damper and S.L MacDonald, "Template Adaptation in Speech Recognition", Proc. IOA, vol. 6, 293-299 (1984).

[10] Y. Niimi and Y. Kobayashi, "Synthesis of Speaker-Adaptive Word Templates by Concatenation of the Monosyllabic Sounds", Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process., April 1986, 2651-2654.

[11] F.R. McInnes, M.A. Jack and J. Laver, "Comparative study of time segmentation and segment representation techniques in a DTW-based word recogniser", IEE Conf. Pub. 258 (Speech Input/Output; Techniques and Applications), 21-26 (1986).

[12] J.S. Bridle, M.D. Brown and R.M. Chamberlain, "An Algorithm for Connected Word Recognition", Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., May 1982, 899-902.

PAPER 3

EXPERIMENTS WITH TEMPLATE ADAPTATION IN AN ISOLATED WORD RECOGNITION SYSTEM

Proceedings of the European Conference on Speech Technology, Edinburgh, September 1987, vol.2, pp.484-497

EXPERIMENTS WITH TEMPLATE ADAPTATION IN AN ISOLATED WORD RECOGNITION SYSTEM

F.R. McInnes*, M.A. Jack*, J. Laver*

ABSTRACT

A template-based isolated word recognition system, with adaptation of templates by weighted averaging with recognised input utterances, is described. Experiments with adaptation of speaker-specific and speaker-independent templates are reported. The results show substantial improvements in the recognition accuracies attained. Aspects of interaction between the system and the user are discussed.

INTRODUCTION

The technique of whole-word template matching (ref 1) for isolated and connected word recognition has attained considerable success and has found practical applications for tasks which involve recognition of words from small to medium-sized vocabularies. The systems available mostly employ speaker-specific templates, formed from utterances of the words by the intended user in a training session. Some success has been attained (ref 2) with speaker-independent systems, using several templates for each word of the vocabulary, formed by clustering from utterances by a standard set of speakers.

A shortcoming of the template-matching approach in its basic form is that the templates are derived entirely from the training utterances provided before the start of a recognition session: no use is made of the additional data acquired during the recognition session in the form of recognised input utterances. An adaptation procedure, by which the initial templates are modified progressively to incorporate information from recognised input, can enhance the performance of a template-based speech recognition system by making the templates more truly representative of the user's pronunciations. This is particularly desirable in a system which starts with speaker-independent templates, as the current user's pronunciations may not correspond closely to any of these templates. Adaptation may also help to track gradual changes in the speaker's voice, during an extended recognition session or over a period of days or months.

An isolated word recognition system incorporating a weighted averaging procedure for adaptation of templates is described briefly below (further details may be found in ref 3), and results are reported which show the effects of this adaptation on the accuracy of recognition. Some issues relating to adaptation and the user-system interface are discussed.

DESCRIPTION OF THE SYSTEM

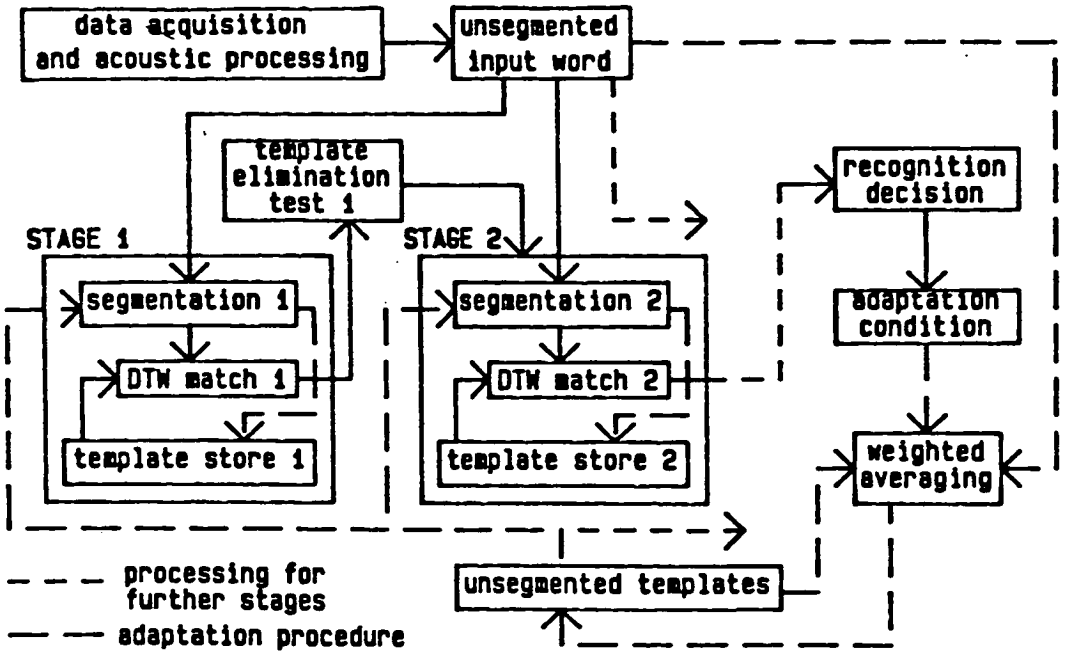
The structure of the isolated word recognition system is illustrated in figure 1. It incorporates a multiple-stage decision procedure, in which successively more detailed comparisons of the input with the templates are carried out until a recognition decision is reached.

For the experiments reported here, three stages were used. Each stage involves division of the input word pattern into a number of equal time segments, and comparison of the resulting normalised pattern with correspondingly segmented forms of the templates by a dynamic programming algorithm (dynamic time warping or DTW) (ref 4). The representation of each word consists of vectors of cepstral coefficients derived from an LPC analysis (ref 5); the segmented form is obtained by averaging these vectors to derive one vector for each segment (or, at the third stage where there are 30 segments per word, interpolating to derive a vector at each segment boundary) (ref 6).

The distances obtained by the DTW comparison at each non-final stage are used to decide which (if any) templates should be matched to the input word at the next stage. If at any comparison stage the ratio of the distances for the best two recognition candidates exceeds a threshold set for that stage, the input is recognised as the word whose template has the smallest distance. Thus the recognition decision may be taken at any of the three stages, depending on whether one word of the vocabulary matches the input much better than any of the others. If the decision can be made after the first stage, the computational cost of the recognition process is very small, as each word is represented at this stage by just two averaged cepstral vectors, and the DTW thus reduces to a very simple linear matching of the input and the template.

*Centre for Speech Technology Research, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN

FIGURE 1: STRUCTURE OF RECOGNITION SYSTEM



Once an input word has been recognised, it may be used to adapt the template which has yielded the smallest distance. There are several possible criteria that may be applied to determine whether to perform adaptation.

If there is explicit feedback from the user as to the correctness of the recognition, the condition can be imposed that the recognition must be correct. This case is referred to as *supervised* adaptation. If the recognition is incorrect, the template may be adapted negatively, away from the input word, to make the recurrence of the same misrecognition less likely. It is also possible to test the second-best candidate, where the best is incorrect, and adapt its template also, positively or negatively as appropriate.

Another form of verification of the recognition is possible if the vocabulary includes the special word "CORRECTION". In this case the adaptation to the most recent input word is delayed until the next input is recognised; if this next word is not identified as "CORRECTION", the preceding recognition is assumed to be correct. The indications of correctness or incorrectness obtained by this means can be used to control template adaptation as in the case with explicit verification. The main disadvantage of this option is that wrong adaptations can occur if the word "CORRECTION" is not recognised reliably.

A third form of adaptation condition, which allows *unsupervised* adaptation, does not rely on having any verification of the recognition by the user. The condition imposed in this case is that the ratio of the best two candidates' distances should exceed a threshold value, set to prevent adaptation in cases where the identification of the input is not sufficiently certain.

The adaptation procedure consists of a weighted averaging with DTW alignment (ref 7) applied to the recognised input word and the template to be adapted. The weights on the input and the existing template can be kept constant at successive adaptations, or they can be adjusted so that the ratio of the template weight to the input weight increases linearly with the number of utterances that have gone into forming the template. The former system of weighting is called the *tracking* formulation, because the contribution of each input utterance to the adapted template decays exponentially with subsequent adaptations and so the form of each template depends mainly on the most recent inputs. The latter system is the *optimisation* formulation. Here weights are assigned according to amounts of data, and so an adapted template contains equally weighted contributions from all input utterances used to adapt it.

To improve the stability of the system when the adaptation is unsupervised, a "skewed" adaptation option is provided, for use when there are several templates for each word of the vocabulary (ref 8). The template adapted to any input utterance is not the template with the smallest distance, but the next template in the list for the same word of the vocabulary.

Besides "CORRECTION", two other special words can be included in the vocabulary: "STOP", which, when recognised, causes the termination of the recognition session; and "RETRAIN", which allows

retraining (i.e. formation of a new template to replace the existing one) for any word or words of the vocabulary (which the user selects by keyboard input).

EXPERIMENTS AND RESULTS

Isolated word recognition experiments with template adaptation have been performed using two data bases, with speaker-specific and speaker-independent initial templates respectively.

The data for the speaker-specific template adaptation experiments consisted of words uttered by one male speaker, collected during interactive adaptive recognition sessions with the system described above. The vocabulary, of 50 words, comprised numbers, days of the week and month names. The training and recognition sessions were conducted in a computer terminal room with a moderate but variable level of background noise, using a headset microphone.

Two initial template sets were used, each containing one template for each word in the vocabulary. In template set T1, each template was formed from a single utterance; in set T2, each template was derived by averaging from two utterances of the word. The test data consisted of 10 repetitions of the vocabulary. The numbers of recognition errors occurring on these 10 repetitions are shown in table 1, for cases with and without adaptation. The tracking form of weights was used. (Similar results were obtained with the optimisation form, except that the result using T2 with unsupervised adaptation was improved to 92.8%). In the supervised adaptation case, negative adaptation was employed in cases of misrecognition, but there was no adaptation of the second-best template.

Table 1 Results with adaptation of speaker-specific templates

Adaptation	Errors on repetitions of 50 words										Overall accuracy
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th	
(T1)											
None	7	11	8	4	6	2	5	8	5	5	87.8%
Supervised	6	9	7	3	3	2	5	5	3	2	90.0%
Unsupervised	7	10	7	6	6	5	6	9	4	3	87.4%
(T2)											
None	5	5	6	4	8	5	4	7	2	4	90.0%
Supervised	6	4	4	2	1	3	2	4	1	2	94.2%
Unsupervised	6	4	6	2	7	4	4	6	2	2	91.4%

For the speaker-independent recognition experiments, the training data consisted of one repetition of the 10 digits by each of 50 training speakers (37 male and 13 female). Results are given here for two sets of templates, the first (D1) containing six templates for each digit, derived by a criterion based exchange clustering procedure (ref 9), and the second (D2) containing two templates per digit, obtained by separate averaging of the utterances of the male and female training speakers. The test data were three repetitions of the digits spoken by each of 49 speakers (37 male and 12 female) who were not in the training set. The words spoken by each test speaker were recognised, with and without adaptation, using each of the two sets of initial templates. Various forms of adaptation, with the optimisation form of weighting, were tested. A word length normalisation (to 30 vectors per word) by linear time segmentation was applied to all the utterances prior to the clustering and recognition processes.

The results, averaged over the 49 test speakers, are shown in table 2. Average recognition accuracies are given for the first, second and third repetitions of the digits by each speaker, and for the whole set of three repetitions, using each set of initial templates.

Line (1) of table 2 shows the results with no adaptation of the templates. The remaining lines show results with adaptation. The number given after "w" in the left column of the table is the ratio of the weights assigned to the initial template and to each input utterance used in template adaptation. The smaller this ratio is, the faster the adaptation. When it is 0, each adapted template is simply the average of the input utterances used to adapt it.

Lines (2) and (3) show results with supervised adaptation, including negative adaptation (with a small negative weight on the input utterance) for misrecognitions, but not including any adaptation of second-best recognition candidates. Lines (4) and (5) give the corresponding results with second-best candidate adaptation allowed. The improvements over line (1) for the third repetitions show the effect of the adaptation to the preceding two repetitions.

Lines (6) to (8) show results with unsupervised adaptation. The results in line (6) are with adaptation of the best-scoring template; those in lines (7) and (8) are with skewed adaptation. With one exception, these results with unsupervised adaptation show decreases in recognition accuracy.

Table 2 Results with speaker-independent initial templates

Adaptation	Template set D1 Input repetitions				Template set D2 Input repetitions			
	1st	2nd	3rd	all	1st	2nd	3rd	all
None								
(1)	92.4%	94.6%	92.0%	92.99%	91.4%	92.7%	90.6%	91.56%
Supervised								
(2) w1	90.2%	93.9%	94.7%	92.92%	87.8%	91.0%	93.1%	90.61%
(3) w0	90.4%	94.3%	95.3%	93.33%	89.4%	91.4%	94.3%	92.52%
(4) w1 +	89.0%	96.3%	97.8%	94.35%	86.5%	95.7%	97.4%	93.13%
(5) w0 +	89.8%	96.1%	97.6%	94.49%	87.8%	95.7%	97.4%	93.54%
Unsupervised								
(6) w4	91.4%	92.4%	91.4%	91.77%	91.0%	90.0%	88.4%	89.80%
(7) w4 skew	92.0%	93.1%	92.0%	92.38%	90.8%	92.0%	89.0%	90.61%
(8) w2 skew	92.0%	91.6%	90.8%	91.77%	91.0%	90.0%	88.4%	89.80%

The recognition of each speaker's first repetition of the digits is consistently poorer with adaptation than without. This occurs because, during recognition of the first repetition of the vocabulary, the template set is a mixture of unadapted and adapted templates; an adapted template for an incorrect candidate recognition may be closer to the input word than the unadapted correct-candidate template, because adapted templates correspond better to the speaker's voice.

DISCUSSION AND CONCLUSIONS

It is evident from the results obtained that supervised adaptation of templates during recognition sessions can significantly improve isolated word recognition accuracy, whether the initial templates are speaker-specific or speaker-independent. Moreover, the improvement is attained more rapidly, at least with speaker-independent initial templates, if adaptation can be applied not only to the best-matching template but also, where the best candidate is incorrect, to the second-best.

The results with unsupervised adaptation are less consistent. It yielded a net improvement in results with speaker-specific templates, but a deterioration with speaker-independent templates - though in the case of skewed adaptation for multiple templates (D1) the results are not conclusive, and experiments with more extended input sequences will be required to determine whether this adaptation is beneficial.

In the system described here, template adaptation improves not only the accuracy but also the speed of recognition, because, when the templates are well tuned to the speaker, fewer comparisons are required at the later stages of the decision procedure. However, the adaptation itself takes some computing time - often more than the actual recognition. This computation could be reduced by using a linear averaging operation instead of the DTW method.

The design of the interaction between the recognition system and the user is important. By including a convenient means for the user to correct wrong recognitions, and delaying the adaptation to each input until an opportunity for such correction has been given, supervised adaptation can be implemented without the need for an explicit yes/no response by the user to each recognition. The facility for retraining templates as required is a desirable feature, particularly if there is any risk of instability arising from adaptation to inputs which are misrecognised or affected by noise.

ACKNOWLEDGEMENT

The work reported here was made possible by an SERC research studentship.

REFERENCES

1. L R Rabiner & S E Levinson, IEEE Trans Commun COM-29, 621 (1981)
2. L R Rabiner, S E Levinson, A E Rosenberg & J G Wilpon, IEEE Trans Acoust, Speech, & Signal Process ASSP-27, 336 (1979)
3. F R McInnes, M A Jack & J Laver, Proc Inst of Acoust 8, 7, 283 (1986)
4. F Itakura, IEEE Trans Acoust, Speech, & Signal Process ASSP-23, 67 (1975)
5. A H Gray & J D Markel, IEEE Trans Acoust, Speech, & Signal Process ASSP-24, 380 (1976)
6. F R McInnes, M A Jack & J Laver, IEE Conf Pub 258 (SIOTA 86), 21 (1986)
7. R Zelinski & F Class, Proc IEEE ICASSP 83, 1053 (1983)
8. T R G Green, S J Payne, D L Morrison & A Shaw, Behaviour & Inf Tech 2, 1, 23 (1983)
9. A Mokeddem, H Hugli & F Pellandini, Proc IEEE-IECEJ-ASJ ICASSP 86, 2691 (1986)

PAPER 4

REFERENCE TEMPLATE ADAPTATION IN SPEAKER-INDEPENDENT ISOLATED WORD SPEECH RECOGNITION

Electronics Letters, vol.23, no.24, 19th November 1987, pp.1304-1305

REFERENCE TEMPLATE ADAPTATION IN SPEAKER-INDEPENDENT ISOLATED WORD SPEECH RECOGNITION

Indexing terms: Signal processing, Speech processing, Speech recognition

A technique which permits the adaptation of reference patterns (templates) in isolated word speech recognition systems is described. Experimental results for supervised and unsupervised adaptation with speaker-independent initial templates are presented.

Introduction: Many automatic speech recognition systems rely on a whole-word template-matching technique, using a dynamic programming algorithm[1] referred to as 'dynamic time warping' (DTW). In a system using this technique, the recognition accuracy depends critically on the relationship between the reference templates and the speech of a specific user.

There are two common approaches to deriving the reference templates for use in such a system. The speaker-dependent approach constructs reference templates by requiring the prospective user to provide at least one utterance of each word in the vocabulary, before starting to use the system. The alternative approach constructs a speaker-independent set of reference templates from utterances by a representative group of training speakers.[2] Here several templates per word are required, to allow for the variations in pronunciation among potential users.

A technique is presented here for use in template-based word recognition systems, which permits ongoing adaptation of speaker-independent reference templates during a recognition session, to take account of the information in the recognised utterances.[3–6] Adaptation can be supervised (conditional on the user's verification of each recognition) or can be unsupervised. Several different adaptation options are defined and compared here. The performance of the template adaptation technique is discussed when it is incorporated in a multiple-stage decision processor[5] for speaker-independent isolated word recognition.

System description: The structure of the recognition system has been described in detail elsewhere.[5,6] It incorporates three stages of word pattern comparison, with progressively more complex representations of the words (derived by a segmentation technique for time compression) at successive stages. This allows some templates to be eliminated with a small amount of computation: a fuller comparison is performed only for the best-matching templates. The recognition decision may be reached at any of the three stages.

When using the supervised mode for adaptation of reference templates, the system determines (by prompting the user) whether the recognition of the word is correct. If the recognition is correct, a weighted averaging procedure (with DTW alignment) is applied to adapt the correct reference template towards the recognised word. Otherwise, an incorrect reference template can be adapted away from the input word, to make recurrence of the error less likely.

In the unsupervised mode, the condition for adaptation is that the ratio of the DTW distances obtained for the best two candidate words (at the stage where the recognition decision is reached) should be greater than a specified threshold. If this condition is satisfied, the recognition is assumed to be reliable, and the weighted averaging is applied; otherwise, no adaptation takes place.

The adaptation here incorporates the 'optimisation' form of weighting:[6] the weights on the existing reference template and on the input word in the weighted averaging procedure are adjusted at successive adaptations so that the relative weight on the template increases linearly with the number of input utterances that have been used to adapt it. After any number of adaptations, each reference template is a weighted average of the initial template and the input utterances used to adapt it, in which all these inputs have equal weight.

Experimental results: Results are presented in Figs. 1 and 2 for recognition of spoken digits using speaker-independent initial templates with supervised and unsupervised adaptation, respectively. The results obtained on the same data without template adaptation are also shown, for comparison.

Speaker-independent initial templates (six templates for each digit from 'zero' to 'nine') were derived by clustering and averaging from 50 utterances of each digit, one by each of 50 training speakers. The test data consisted of three repetitions of each digit by each member of a (different) group of 49 test speakers. Because the behaviour of the system with adaptation depends on the order of the input words, the results given are averaged over four different random orderings of the 30 digits from each test speaker.

Fig. 1 shows the effects of supervised adaptation. Average recognition accuracies, with and without adaptation, are

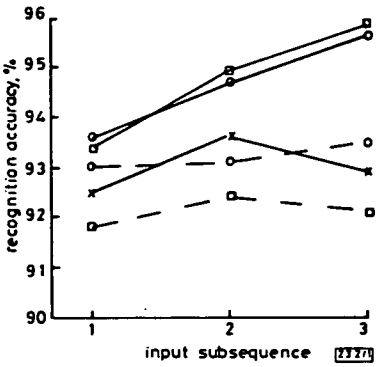


Fig. 1 *Digit recognition results with supervised adaptation*

Adaptation:
× none
□ template weight = 1·0
○ template weight = 0

Compensation:
---- none
——— optimal

plotted for the first, second and third 10-word subsequences of each sequence of 30 digits. Results are shown for two different weighting options. In the first weighting option (identified as 'template weight = 1·0'), when a template is adapted for the first time, the initial template and the input utterance are given equal weights. At the nth adaptation the weights on the template and on the input are adjusted to be in the ratio $n:1{\cdot}0$, so that the adapted template is the average of the initial template and all the inputs used to adapt it. In the second weighting option ('template weight = 0'), the initial template is given a weight of 0, so that the weighted averaging in the first adaptation of a template reduces to a simple replacement of the original speaker-independent template by the recognised input utterance. Here, at the nth adaptation, the weights on the template and on the input are adjusted to be in the ratio $(n-1):1{\cdot}0$. In each case, negative adaptation was applied in cases of misrecognition.

It has been shown previously[6] that recognition errors occur with template adaptation because, when some but not all of the templates have been adapted, the adapted templates correspond more closely to the speaker's voice characteristics than the unadapted ones, and so an adapted template for an incorrect candidate recognition can be closer to the input than an unadapted correct-candidate template. To compensate for this effect, an adjustment of the DTW distances has been introduced here, whereby each distance is multiplied by a quantity which increases with the number of times the template has been adapted. The results plotted with broken lines in Fig. 1 are without this compensation, while those plotted with solid lines are results obtained using heuristically optimised compensation factors. Without compensation, no significant improvement in recognition accuracy was obtained over the results without adaptation. With compensation, however, improvements of 2·9% and 2·7% over the performance without adaptation can be seen for the last 10 words of each input sequence, revealing the beneficial effect of the previous adaptation to the first 20 words.

Reprinted from ELECTRONICS LETTERS 19th November 1987 Vol. 23 No. 24 pp. 1304-1305

Fig. 2 shows similar results with unsupervised adaptation. Here, at the nth adaptation the template and input weights

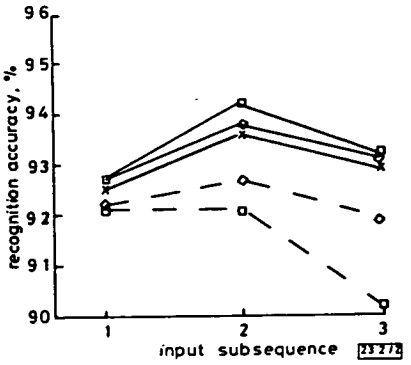


Fig. 2 *Digit recognition results with unsupervised adaptation*

Adaptation:
× none
□ template weight = 4·0
◇ template weight = 4·0, skewed

Compensation:
---- none
——— optimal

are adjusted to be in the ratio $(n + 3{\cdot}0) : 1{\cdot}0$, to help ensure stability of the system. Again, results without compensation and with optimised compensation are given. Without compensation, instabilities occurred, as some templates were repeatedly adapted towards wrongly recognised input utterances. A distinction is made in Fig. 2 between straightforward adaptation of the best-matching template and a 'skewed' form of adaptation. In the skewed adaptation, designed to improve the stability of the unsupervised system, the template adapted is not that which yields the smallest distance, but the next template in the list for the same word of the vocabulary.[3] It can be seen that, without compensation, the instability effect was lessened by the use of skewed adaptation, but, when appropriate compensation factors were applied, the direct form of adaptation yielded marginally greater average improvements than the skewed form.

Conclusions: It has been demonstrated that speaker-independent template-based word recognition performance can be progressively improved by supervised adaptation. Less significant improvements have been observed with unsupervised adaptation. The importance of applying compensation factors, to prevent errors arising from the uneven progress of the adaptation across the different words of the vocabulary, has been demonstrated. In these experiments no benefit was observed to accrue from using a more sophisticated adaptation strategy ('skewed' adaptation) to improve the system's stability in the unsupervised case.

Supervised adaptation need not require explicit verification of each recognition by the user of the recogniser: the 'recognition correct' signal can consist of the absence of an attempt by the user to correct the recognition before proceeding to the next word of input. Thus the benefits of supervised adaptation can be obtained without making much extra demand on the user, if the interface to the recognition system includes a means of correcting misrecognitions.

Acknowledgment: The work reported here was supported by an SERC research studentship.

F. R. McINNES
M. A. JACK

Centre for Speech Technology Research
University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN, United Kingdom

16th October 1987

References

1 RABINER, L. R., and LEVINSON, S. E.: 'Isolated and connected word recognition—theory and selected applications', *IEEE Trans.*, 1981, COM-29, pp. 621–659
2 RABINER, L. R., LEVINSON, S. E., ROSENBERG, A. E., and WILPON, J. G.: 'Speaker-independent recognition of isolated words using clustering techniques', *ibid.*, 1979, ASSP-27, pp. 336–349
3 GREEN, T. R. G., PAYNE, S. J., MORRISON, D. L., and SHAW, A.: 'Friendly interfacing to simple speech recognizers', *Behav. & Inf. Technol.*, 1983, 2, pp. 23–38
4 DAMPER, R. I., and MACDONALD, S. L.: 'Template adaptation in speech recognition', *Proc. Inst. Acoust.*, 1984, 6, pp. 293–299
5 MCINNES, F. R., JACK, M. A., and LAVER, J.: 'An isolated word recognition system with progressive adaptation of templates', *ibid.*, 1986, 8, pp. 283–290
6 MCINNES, F. R., JACK, M. A., and LAVER, J.: 'Experiments with template adaptation in an isolated word recognition system'. Proc. European conf. on speech technology, 1987, 2, pp. 484–487