Journal of Phonetics (1981) 9, 197–220

ERIS-context sensitive coding in speech perception

Stephen Michael Marcus
Institute for Perception Research (IPO), Den Dolech 2, Eindhoven, The Netherlands

Received 1st *March* 1980

Abstract: A number of problems in speech recognition arise through the treatment of speech as a linear temporal sequence. These include word onset detection, temporal normalisation and variations in pronunciation. It is suggested, following Wickelgren (1969, 1972) that speech should instead be represented by a non-sequential *associative* or *context-sensitive* code.

ERIS, a computer speech recogniser based on a set of independent context-sensitive coded demons, demonstrates the validity and power of such an approach. Ways of incorporating absolute time information into a context-sensitive code are discussed, together with the possible need for and nature of intermediate levels of processing between the acoustic stimulus and a word or morpheme representation. Rather than postulating any such units in advance, it is suggested that by considering word recognition as an acoustic–lexical mapping, it will become apparent what intermediate levels are either necessary or useful.

The power of even a relatively simple recognition system based on context sensitive coding and direct acoustic–lexical mapping suggests that these are important principles to be considered in any approach to understanding, modelling and simulating human speech perception.

Introduction

The speech signal reaches the ear of the listener as a pressure waveform varying with time. This signal may be analysed to give units of various size and complexity, varying from samples of the acoustic waveform itself, through sets of acoustic parameters, to phonemes, syllables, morphemes and words, and finally to phrases, sentences and whole units of discourse. Whatever units are ultimately chosen, the problem of speech recognition is one of mapping an unknown input sequence onto properties of a set of stored, previously encountered, representations: in the case of words, the listener's vocabulary.

It is convenient to analyse and represent the speech stimulus as a temporally ordered sequence, and the stored representations are generally coded in a similar manner. Many of the current problems in speech understanding are then concerned with optimal comparison between two such linear representations.

This paper follows Wickelgren (1969) in proposing a non-sequential code for storing and comparing linear temporal sequences The viability of such a coding is demonstrated by ERIS, a simple computer implementation using real speech data. Both for simplicity, and since it is the subject matter of the experimental work presented, I shall restrict myself in

0095–4470/81/020197 + 24 $02.00/0

the following discussion to word recognition, though equivalent problems arise without this restriction.

A fundamental aspect of the current approach is the belief that both psychological models of human speech perception and attempts at machine speech recognition give complementary insights into the same central phenomena. The path likely to give us the deepest understanding of these phenomena is thus one synthesising both sources of knowledge. A hypothesis of Optimal Adaptation, which takes note that until recently speech has been almost exclusively the province of the human speaker–listener, is proposed to guide us in this synthesis.

Onset detection

The initial problem in word recognition is just that – the detection of the onsets of words. We perceive speech as apparently consisting of a sequence of distinct words, but even casual inspection of the acoustic waveform reveals no such neat packages. Figure 1 illustrates this with the amplitude–time representation of a sequence that an English listening listener would perceive as a number of repetitions of the digit "six". The only acoustic silence is for the closure of the stop consonant /k/ in the *middle* of each word.

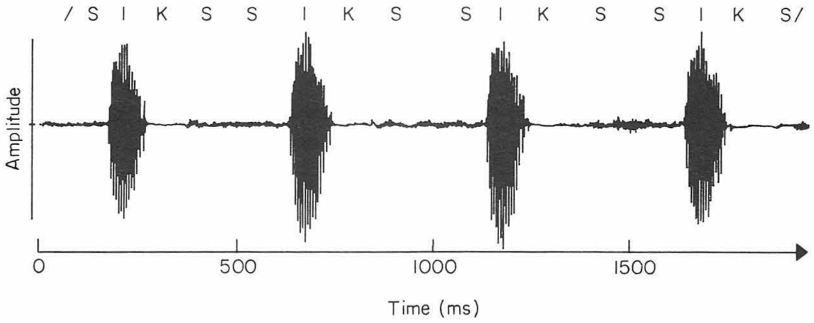

Figure 1 Word boundaries and acoustic silences do not correspond in the sequence "six, six, six, . . . ".

This problem is far from trivial, and it almost appears to be necessary to recognise a word before it is possible to know where it begins. Indeed, with sufficiently unlikely constructions, this may clearly be shown to be the case, as demonstrated by the following examples:

1. "Maaien abten hooi?" (Dutch – "Do abbots mow hay?")
2. "In coal none is." (after Reddy, 1976)

Given such perfectly valid sequences, most listeners, especially with the Dutch example, completely fail to establish word boundaries unless they know in advance what the words in the sentence are. It is even possible to construct sequences in which the speaker and listener have different word boundary positions, the speaker's due to his prior knowledge of one possible segmentation, and the listener's because his linguistic experience provides a more likely segmentation:

3. "Well oiled bead hammed."

Even when the speaker utters single isolated words, there is the problem of distinguishing the acoustic signal associated with the word itself from background noise or any hesitations accompanying its production. Needless to say, identification of word offsets is at least as difficult as onsets, and often these are even more poorly acoustically represented.

Variability of production

Time normalisation

A second problem is that no two naturally spoken versions of a word are precisely identical. I shall return later to acoustic and articulatory variations between utterances which result in spectral or phonetic differences. For now let us consider a more basic dimension of difference, rate of utterance, with corresponding changes in speech timing. These changes are not linear with speech rate, certain segments, such as vowels, tending to be more flexible in duration. I shall also ignore the fact that even such duration changes are not "pure" and are accompanied by spectral changes; the formant frequencies of vowels, for example, change in quality to a more neutral value with increasing speech rate (Lindblom, 1963).

Like all idiosyncracies of human behaviour which form the rich variety of life, many changes in segment duration are governed by no simple rules, and each instance needs to be considered in its uniqueness. If we were to know the precise onset and offset of the to-be-recognised word, various segmentation and normalisation procedures could be applied in matching it to each of the stored representations; such a procedure is in fact used in some isolated word recognition systems. However, we have noted that neither onsets nor offsets are readily available in continuous speech. Furthermore, if we do determine the onset and offset of a word and remove it (by tape splicing or more sophisticated digital means) from the context in which it appears, it becomes considerably less intelligible (Miller, Heise & Lichten, 1951), even though word onset and offset are much more clearly defined. This has led the engineers of many automatic speech recognition systems to concentrate on modelling syntactic and semantic processes, using little more than a highly degraded acoustic representation. Powerful techniques have nevertheless been developed for time-warp mapping between an unknown input and a stored representation to allow for non-linear changes in speech rate. Dynamic programming (Forney, 1973) finds the best matching path for a particular stored representation starting at any point in the unknown input. This matching process produces an error score, and matching continues until this error exceeds a permissible maximum or the end of the stored representation is reached. Since this procedure may, in principle, be repeated for all possible starting points in the unknown input, it is not essential to know this starting point in advance. In practice, such uncertainty gives rise to a vast increase in the already high computational load, and some limitations need to be imposed on the extent of this repeated searching. More sophisticated techniques can allow for pronunciation variation by building up the stored representations in the form of transition networks with associated probabilities, in place of simple linear sequences (e.g. Bakis, 1974). Given the rapid increase in speed and complexity of present and future hardware, even the further additional processing load of such approaches should not be considered prohibitive.

Hypothesis of Optimal Adaptation

Speech has evolved as Man's principle means of communication, and remains of tremendous importance even in this modern age of the printed word and electronic data media. Since the development of speech appears to have gone hand-in-hand with Man's evolution from his origins as the mere Naked Ape, we might expect considerable effort to have been devoted to the perfection of this tool. I wish to propose a *Hypothesis of Optimal Adaptation* to guide us in modelling the speech perception process:

> The speech signal is optimally adapted to
> human speech recognition, and vice-versa.

A corollary is that an optimised speech recogniser should have characteristics similar to those found in human speech recognition. Psychological and psycholinguistic data thus outline the performance characteristics of such a recogniser. Current psycholinguistic work which is demonstrating many facets of the real-time performance of human speech recognition is of particular interest.

A distinction originating from the world of artificial intelligence approaches to pattern recognition is that made between "top-down" and "bottom-up" analysis. These correspond to "active" and "passive" recognition theories, so sharply contrasted in psychological theories of the previous decade (see Morton & Broadbent, 1967; Neisser, 1967). "Bottom-up" information originates from the stimulus, and, as it undergoes more and more refined processing, rises towards the "top". "Top-down" information originates from semantic, syntactic and pragmatic constraints, and travels "down" to aid in the interpretation of the stimulus input.

Let us first note that speech recognition may, in extreme cases, exhibit either principally "top-down" or "bottom-up" characteristics. An example of the former would be the perception of the word "trees" in the context "Apples grow on trees". We can test the hypothesis that most subjects would respond "trees" even without the stimulus by asking them to complete the phrase "Apples grow on ——." with one word. If we find that a large percentage of subjects complete this sentence with the single word "trees", we may feel justified in concluding that by the fourth word in the sentence, "top-down" information very strongly favours a particular type of green object.

In contrast, an example of a principally "bottom-up" process is the recognition of an unknown word presented in isolation. We know from common experience that if we speak clearly, such a word can be successfully recognised. Even this is not totally "bottom-up" since it of course relies on the common stored lexical knowledge of speaker and listener.

Lieberman (1963) not only demonstrated that a word clearly identifiable when presented in continuous speech is much less intelligible in isolation, but also a continuum from the recognition of isolated words to the recognition of highly predictable words in continuous speech. If a word is uttered in two alternative contexts, say "nine" in "a stitch in time saves ____", and a much less predictive context such as "the number is ____", although the word "nine" is highly intelligible in both cases, if it is spliced out of these contexts and presented in isolation, the token from the *less* predictable context is much *more* intelligible. This demonstrates not only that human speech perception can optimally combine "top-down" and "bottom-up" information, but also that such optimisation is modelled in the process of speech production. That is, it is "known" just how much care *needs* to be taken in uttering each word in a particular context in order for it to be successfully perceived, and thus to communicate the meaning of a sentence.

Recent experiments by Marslen–Wilson and his associates have provided much information on the real-time nature of this information exchange between acoustic analysis and syntactic and semantic constraints. In his earliest experiments, Marslen-Wilson (1973) studied the so-called "close shadowers". Although these subjects were capable of shadowing speech with latencies as small as 250 ms, Marslen-Wilson demonstrated that semantic and syntactic constraints were operating as effectively as in the normal, "slow", shadowers.

Since the close shadower's latencies were of the order of one or two syllables, the operation of such constraints, which require word identification, demonstrated that it is possible for words to be identified long before the end of their associated stimuli.

Such a phenomenon can also be observed in the phoneme monitoring reaction times of Morton & Long (1976). They were interested in the effect of the transitional probability, that is the "top-down" likelihood, of a word in a given context on the speed of detection of an initial phoneme in that word. They found faster reaction times for initial phonemes in high transitional probability words, and this requires that the word be recognised prior to the production of the phoneme monitoring response. In addition, allowing time for response initiation, the reaction times they recorded required that word recognition must have occurred well before word offset.

These results demonstrate that, given appropriate "top-down" information, the human speech recognition system may initiate word responses very early in the stimulus word. Note that this rules out any recognition system which requires the end of the stimulus word in order to perform some time normalisation process prior to recognition. The relative unimportance of information late in the word is even more clearly demonstrated in a subsequent experiment of Marslen-Wilson's in which subjects were required to shadow material containing mispronunciations (Marslen-Wilson & Welsh, 1978). They found that, although the mispronunciations were easily detectable when that was the subjects' primary task, when subjects were simply asked to shadow the same material, many of the mispronunciations were "restored" to their original form. These "restorations" were most likely when the word containing the mispronunciation had a high transitional probability, and the mispronunciation itself was later in the word. In shadowing, 43% of three feature mispronunciations in the final syllable of high probability words were found to be restored. When detecting mispronunciations, only 3% of these same mispronunciations failed to be detected when that was the subjects' primary task. We may term this restoration "hyperaccurate" perception, and Marslen-Wilson and Welsh found that in these cases there was no perturbation of shadowing latencies for the restored words; it was as if the shadowers restored the words to their original form because they had not noticed the mispronunciation. Only in cases where the mispronunciation was literally repeated was there a noticeable, and quite dramatic, increase in shadowing latencies.

It should be noted that these results also provide no evidence for, and cast some doubt on, the viability of the phoneme as an intermediate percept in speech recognition. If phonemes were to serve such an intermediate function, it should be expected that latency to detect a word initial phoneme would not be dependent on factors influencing the detection of the whole word. Although it could be argued, and has been for example by Rubin, Turvey & van Gelder (1976), that in both cases the phoneme is detected with equal speed and accuracy, and variations in difficulty of the word recognition process interfere with the production of the phoneme detection *response*, this does not constitute evidence *for* the existence of phonemes as intermediate perceptual units. At most, it could be used to argue that such data cannot definitely disprove their existence. It is also useful to remember that phonemes were first postulated as distinguishing between words, or between words and non-words:

> If phonemes are percepts to the native speakers of the language, they are not necessarily percepts that he experiences in isolation. They occur ordinarily as the elements of words or sentences. Phonemes are perceptive units in the sense that the native can recognize as different, words different as to one of the component phonemes. (Swadesh, 1934)

Possibly the most convincing data against a categorical intermediate phonemic level in word recognition comes from a recent paper by Streeter & Nigro (1979). In a speeded lexical decision task on stimuli containing VCV sequences, they examined the effect of either omitting or substituting an incompatible VC transition. Although the CV transition dominated perception of the stop consonant C, they found that incompatible VC transitions slowed down responses to word stimuli. In contrast there was no effect of any of the experimental manipulations on producing non-word responses to non-word stimuli. This result would be difficult to account for if an intermediate phonemic level were required before lexical access (or failure of lexical access). Incompatible VC transitions should then exert an influence on phoneme identification, and consequently on both word and non-word responses.

Real time processing and left-to-right continuity

Despite the evidence presented above for the extremely rapid real-time nature of the human speech recognition process and the greater perceptual salience of segments early in the word, a clear example that speech perception does not require strict left-to-right continuity in the speech signal is provided by the phoneme restoration effect (Warren, 1968; Warren & Gregory, 1958). If a small segment of a word in continuous speech is replaced by noise, the listener not only has the impression that he has perceived the whole word, but is also very poor at localising where the noise occurred in the word. Thus a "hole" in the information associated with a word does not necessarily drastically interrupt its perception, as might be expected from models in which strict left-to-right sequential probabilities between segments are used for recognition and temporal normalisation, as in dynamic programming. More complex approaches could be envisioned in which parallel processors for each possible word each attempt to build "bridges" over any non-match or "hole", following all possibilities to the last syllable of the uttered word. However, recent data on the "Tip of the Tongue" (TOT) phenomenon, in which the phonological coding of words is only partly retrieved, show that in this case segments which are successfully accessed often fail to retain their correct serial order (Browman, 1978). This casts further doubt on a strict serial representation of speech, and the next section will investigate an alternative approach which may offer an elegantly simple solution to a number of these problems.

Context-sensitive coding

Let us represent consecutive samples of a speech signal by the sequence a, b, c, . . . , as shown in Fig. 2. Each element in this sequence represents a sampled state of the input as a point in some parameter space. It is immaterial for our present purposes whether this space is a simple set of labels, such as phonemes, or exhibits considerable dimensional structure and complexity, such as formant and bandwidth data. For generality I shall term the elements *state vectors* without specifying their duration or dimensionality.

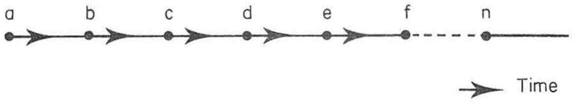

Figure 2 Consecutive samples of a speech signal.

Such a sequence can be, and generally is, represented by storing the elements (or a transformation of them) in the order encountered. Some of the problems associated with using such a linear representation in recognition have been outlined above. An alternative way to represent such information is by a *context-sensitive*, or *associative*, code, in which each element is stored separately with information about the neighbouring elements with which it appears in context.

Order of complexity

Context-sensitive codes may be of any *order of complexity*, including in the coding of each element as many of the neighbouring elements as desired. The simplest possible may be termed a *first-order* context-sensitive code, and it consists of each element plus information on the following (or preceding) element only. Wickelgren (1972) proposed that context-sensitive codes be used for all forms of memory representation other than very short-term sensory buffers, explicitly dealing with segments down to phoneme size and duration (1969). This paper will investigate and demonstrate the viability and power of such a coding for speech recognition using elements of much shorter duration, and suggests that, for speech at least, context-sensitive associative coding may play a major role at all levels in the recognition process.

In a first-order context-sensitive code, the sequence of state vectors illustrated in Fig. 2 would be represented by the *unordered* set of *state-pairs* shown in Fig. 3. Sequential order is not explicitly coded, but may be recovered from the stored code as indicated by the dotted line. In this example the original order may be uniquely reconstructed. This will not always be the case, and it is clear that when some states are repeated in the original sequence, ambiguities may lead to repetition, omission or reversals of segments. Wickelgren (1969) assumed that appropriate state-vectors would be phonemes, and, because of the difficulty in reproducing the correct sequence from an unordered set of phoneme pairs, proposed a second-order context-sensitive code for speech production; in this each phoneme was labelled with both the preceding and the following phoneme, and he termed these triads *context-sensitive allophones*. However it will be argued that in recognition a first-order code has advantages of simplicity, and of greater flexibility along the time dimension.

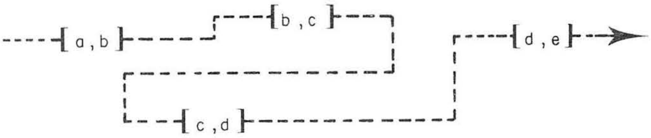

Figure 3 Context-sensitive coding for the sequence shown in Fig. 2. The sequence indicated is not explicitly represented.

Context-sensitive coding in recognition

In recognition, an associative code offers some clear and elegant advantages over a linear code when comparing incoming elements of an unknown stimulus with a stored pattern. Since each input state-pair may be independently mapped onto a stored representation, neither the location of the start point of a word nor subsequent sequential tracking is necessary. The strength of mapping onto each stored representation may be computed and accumulated and the stimulus "recognised" when it matches sufficiently well one of the stored representations rather than any other. Both the required goodness of this match and how much better it need be than all other matches can be varied to give a trade-off between speed and accuracy of recognition. If sections of the stimulus are absent or distorted, either at the beginning of a word or later on, then there will be less information favouring the corresponding stored representation, and less discriminating it from others, but the recognition process is not crucially dependent either on the detection, or even presence, of the start of a word, or of any other segment. However, if segments are distorted and match another representation, this may be generated as a response instead. Furthermore, if there is no stored representation corresponding to the stimulus, one giving a sufficiently good match may be selected as a response. This potential for making "erroneous" responses to distorted or unknown stimuli should not necessarily be considered a drawback for a recognition device. It may be this very property of human speech recognition which allows the listener to make such a good best of a bad job when listening to speech under very noisy conditions. The critical point is whether, given a sufficiently clear signal, an ideal implementation is able to select the correct representation with the same reliability and real-time characteristics as the human listener.

Variability of pronunciation

Each token utterance of a nominally identical word will normally differ in some aspects from any other token. These differences will be greater for different speakers, but even for the same speaker one token cannot be taken as truly representative of his pronunciation of that word. Given a number of tokens of a word produced by the same speaker [Fig. 4(a)], there will generally be a number of points at which there are common state-vectors, and with these points as nodes, a state transition network may be built up for each word [Fig. 4(b)]. The empirically determined probabilities of transitions between states may be incorporated, and the recognition process involves finding the optimum path through the network corresponding to an unknown stimulus. The network having the highest probability optimum path dictates the final response. A number of systems have been implemented on this principle, examples of which are the HARPY system (Lowerre, 1976) and Bakis' "word spotter" (Bakis, 1974).

Context sensitive codes for the tokens in Fig. 4(a) will each consist of an unordered set of state-pairs. Some state-pairs will be common to more than one token, and these will correspond to shared paths in the state transition network. If we combine all state-pairs from all these tokens, we have a collection onto which each of the original tokens will match, but now there is considerably more ambiguity in recovering any of the original sequences. In fact we may now assume that reconstruction of something approximating any of the original orderings has become effectively impossible. However, this extra ambiguity results from combining information on the variability of pronunciation of a particular word, and it may be that we are building up just what we require – a representation broad and flexible enough to contain and delimit all alternative pronunciations of a given word.

In dealing with variations in speech rate, we see a particular advantage of the first-order code suggested here over the second-order code employed by Wickelgren (1969). If the sequence #abc is produced as #abbc, that is with a temporal extension of one segment, the corresponding context-sensitive codes will be:

Stimulus:	#abc	#abbc
First-order:	[#a], [ab], [bc]	[#a], [ab], [bb], [bc]
Second-order:	[#ab], [abc]	[# ab], [abb], [bbc]

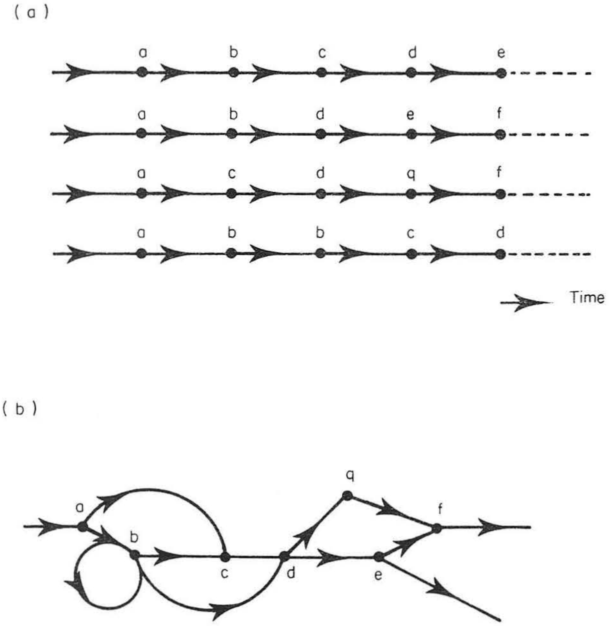

Figure 4 (a) Sampled states in a number of token productions of a word. (b) A state transition network for the tokens in (a).

The codes in the dyads or triads which match between the two stimuli have been underlined, and it can be seen that the second-order code is relatively inflexible in dealing with variations along the time dimension. In fact, Wickelgren's arguments in favour of a second-order over a first-order code could be reversed, and it might be suggested that what appears as *excessive ambiguity* in production should be thought of as *desirable flexibility* in perception.

A cautionary note

By discarding all temporal information other than element-to-element local context, this coding probably fails to retain certain temporal information relevant in speech perception. The nature, importance and extent of such omission, and the way in which it could be re-incorporated, are open to investigation. It may be that the high redundancy of speech, in the information theoretical sense (Shannon, 1948), allows a considerable relaxation of sequential constraints, many possible ambiguities simply not being available as legal responses in the language.

In order to empirically investigate the power and limitations of a first-order context-sensitive code in speech recognition, a computer simulation was implemented working on real speech parameters. The ERIS program and its results are described in the following sections.

ERIS I – a computer simulation

ERIS is an implementation of the first-order context-sensitive code outlined in the last section. Despite its simplicity, it provides a practical demonstration of the power of such a coding as a general principle in speech recognition. Hardware limitations and a desire not to initially become too deeply involved in multidimensional mapping procedures led to the use of a coarse acoustic representation and a simple one-dimensional stimulus space.

Parameters

The basic input data were speech formant parameters extracted by the IPO linear-predictor coefficient (LPC) formant vocoder system (Vogten & Willems, 1977). The first three formants were chosen rather than any other equally arbitrary set of spectral descriptors, such as the LPC coefficients themselves or a power spectrum, because a not inconsiderable amount is known about the psychological and phonetic importance of formant values and formant transitions (see e.g. Lindblom, 1963; Liberman, Cooper, Shankweiler & Studdert-Kennedy, 1967; Fant, 1969). Additionally, attempts at speech synthesis in general, and the use of the formant vocoder in particular, have demonstrated that intelligible speech may be produced from such an oversimplified description of the speech signal. Not only formant frequencies but also changes in formant frequencies have a systematic influence on speech perception; therefore, rather than coding state-pairs as $[\underline{a}, \underline{b}]$, where $\underline{a}$ and $\underline{b}$ represent consecutive state-vectors, an equivalent representation $[\underline{a}, \underline{\dot{a}}]$ was chosen, where $\underline{\dot{a}} = \underline{b} - \underline{a}$. This code has the advantage both of directly representing changes in formant frequency and of being more economical when it comes to mundane practicalities of computer storage. This second advantage comes from the relatively slow variation of the speech parameters, parameter changes being generally an order of magnitude smaller than their absolute values. In contrast to static spectral templates often used in automatic speech recognition, such state-pairs contain both steady-state and changing parametric information. When using a limited set of them to describe the speech signal, we may thus term them *dynamic spectral templates*.

Unfortunately, even these simplest formant and bandwidth parameters, together with some basic information on voicing and amplitude, give rise to far too many possible state-pair values to be handled on a large computer, let alone the modest minicomputer on which ERIS runs (a 16 bit Philips P9202 with 16 k core memory and 4 M words disc). To restrict the possible combinations, a subset of the parameters was quantised as shown in Table I.

Table I Order and number of bits assigned to ERIS parameters

Parameter				Bits
1	V	voicing	voiced/unvoiced	1
2	A_0	amplitude	4 values	2
3	F_1	formant 1	4 values	2
4	F_2	formant 2	8 values	3
5	F_3	formant 3	4 values	2
6	B_1	bandwidth F_1	wide/narrow	1
7	DF_1	change F_1	rising	1.58 (= 2)
8	DF_2	change F_2	stationary	1.58 (= 2)
9	DF_3	change F_3	falling	1.58 (= 2)
10	DA_0	change A_0	onset/continue/offset	1.58 (= 2)

Figure 5 shows the LPC-formant vocoder parameters for the digits "one", "two" and "three" before and after quantisation. Values of the five formants and amplitude are given for each 10 ms frame. The height of each vertical stripe indicates the *Q*-factor of that formant.

Speech processed in this way neither sounds very pleasant nor very clear when re-synthesised; there remain nevertheless 165888 possible state-pairs. It was hoped that a much smaller number would actually be encountered in a sample of real speech.

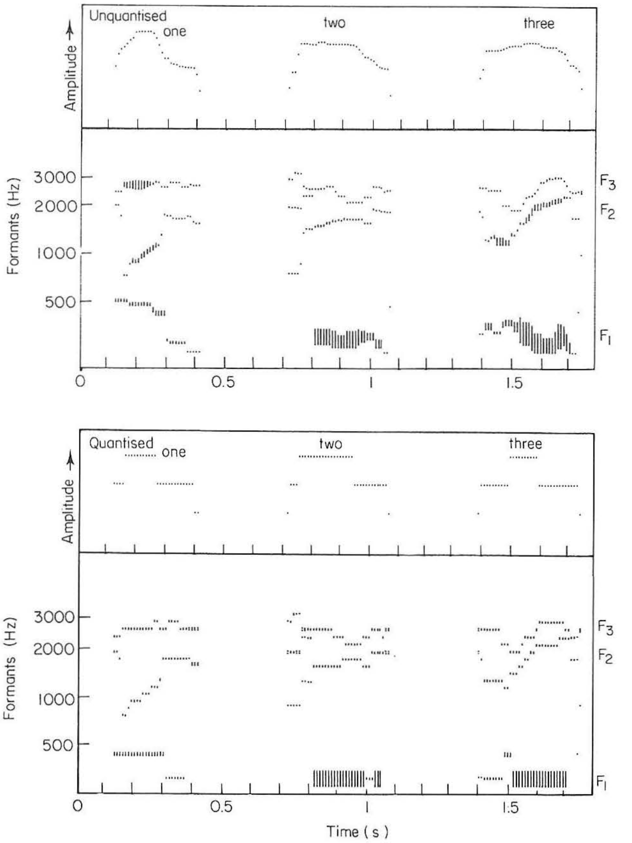

Figure 5 Tokens of the digits "one", "two" and "three" illustrating unquantised and quantised amplitude and formant parameters.

Having no simple multidimensional structure for the state-pair descriptor shown in Table I, the bits allocated to each parameter were assembled in the order shown, parameter 1 being the most significant, into a 19 bit unsigned integer, and the value of this integer was taken as a value on a one-dimensional scale. This value defines the one-dimensional *state-pair vector*, **S**.

No intermediate stage was introduced between these 10 ms state-pairs and whole word recognition. In particular, no level corresponding to a possible phonetic or phonemic code was employed. Since there is no convincing evidence that such units play a role in human speech recognition, following the hypothesis of Optimal Adaptation, there is no reason to suppose that they are required in an optimised recogniser. It was expected that if such levels of intermediate representation are in fact useful or necessary, this would become evident from the implementation itself. Thus, by not making any *a priori* assumptions we may learn a considerable amount more about what really lies between the acoustic signal and word recognition. The choice of 10 ms sample frames is of course arbitrary, and no magic properties are associated with this interval. We know from experience with speech synthesisers that a frame rate of this order is able to follow most of the significant changes in the speech signal. Faster rates may be considered an elegant luxury, and slower rates risk losing some transient burst information, though work with the IPO formant vocoder has demonstrated good results with frame durations as long as 30 ms (Vogten & Willems, 1977).

Speech corpus

The English digits "one" to "nine" were arbitrarily chosen to form a small lexicon for all the simulations. A number of tokens of each digit were spoken by the author (a native speaker of English) and analysed with the LPC-formant vocoder system. The formant data were then quantised and reduced as shown in Table I. One token of each digit was chosen at random and retained for the subsequent recognition phase; the rest were used in training. Details of the corpus are given in Table II. Some state-pairs occur either more than once in the same token or in other tokens, and the 4367 state-pairs in the training corpus consist of 1278 different state-pairs.

Table II Composition of speech corpus

Digit	Training		Recognition token
	Tokens	State-pairs	State-pairs
"one"	12	434	30
"two"	8	304	38
"three"	11	436	30
"four"	11	383	29
"five"	13	514	21
"six"	13	622	28
"seven"	15	671	28
"eight"	11	340	28
"nine"	11	663	52
Total	105	4367	274

The number of training tokens of each digit and the total number of state-pairs they contain are given, together with the number of state-pairs in the single recognition token.

A demon

ERIS is based on a collection of independent word *demons*. Each of these is responsible for the recognition of one particular word, and the rejection of all other stimuli as "non-words". During training of any one demon, all tokens of its own words are presented as "words", and all other tokens as "non-words". For example, the "one" demon is presented with the 434 state-pair vectors from the 12 tokens of "one" and told that these are "words", and with the 3933 state-pair vectors from the 93 tokens of "two" to "nine" and told these are "non-words". In the case of the digit corpus used here, all stimuli were thus in fact words, and any particular token's status as "word" or "non-word" varies as it is presented in training different demons. A demon keeps count of how many times each state-pair vector has occurred in a token of its "word", and how many times in a "non-word". For state-pair vector $\mathbf{S}_i$ in demon $\mathbf{x}$, let us term these frequency counts w_{xi} and n_{xi} respectively. Let $w_{x.}$ and $n_{x.}$ similarly denote the *total* number of "word" and "non-word" state-pair vectors presented to demon $\mathbf{x}$. Let X and N indicate the occurrence of a "word" and a "non-word" for demon $\mathbf{x}$, respectively. We have thus:

$$P(\mathbf{S}_i \mid X) = w_{xi}/w_{x.}, \tag{1}$$

$$P(\mathbf{S}_i \mid N) = n_{xi}/n_{x.}, \tag{2}$$

also

$$P(N) = 1 - P(X), \tag{3}$$

and

$$P(\mathbf{S}_i) = P(\mathbf{S}_i \mid X)P(X) + P(\mathbf{S}_i \mid N)P(N). \tag{4}$$

We can therefore derive the probability, $P(X \mid \mathbf{S}_i)$, that an occurrence of $\mathbf{S}_i$ in an unknown stimulus is a sample from a "word" for demon $\mathbf{x}$:

since

$$P(X \mid \mathbf{S}_i)P(\mathbf{S}_i) = P(\mathbf{S}_i \mid X)P(X), \tag{5}$$

$$P(X \mid \mathbf{S}_i) = \frac{w_{xi}}{w_{x.}} \frac{P(N)}{(w_{xi}/w_{x.})P(X) + (n_{xi}/n_{x.})P(N)}$$

$$= \frac{w_{xi}n_{x.}P(X)}{w_{xi}n_{x.}P(X) + n_{xi}w_{x.}[1 - P(X)]}, \tag{6}$$

and

$$1 - P(X \mid \mathbf{S}_i) = \frac{n_{xi}w_{x.}[1 - P(X)]}{w_{xi}n_{x.}P(X) + n_{xi}w_{x.}[1 - P(X)]}, \tag{7}$$

thus

$$\frac{P(X \mid \mathbf{S}_i)}{1 - P(X \mid \mathbf{S}_i)} = \frac{w_{xi}n_{x.}P(X)}{n_{xi}w_{x.}[1 - P(X)]}, \tag{8}$$

taking logarithms:

$$\text{logit}\ [P(X \mid \mathbf{S}_i)] = \log\,(w_{xi}n_{x.}/n_{xi}w_{x.}) + \text{logit}\ [P(X)] \tag{9}$$

where

$$\text{logit}\ (Z) = \log\,[Z/(1 - Z)]$$

let

$$L(X \mid \mathbf{S}_i) = \text{logit}\ [P(X \mid \mathbf{S}_i)] \tag{10}$$

assuming that the contribution of each frame is independent:

$$L(X \mid \mathbf{S}_i \cap \mathbf{S}_j) = L(X \mid \mathbf{S}_i) + L(X \mid \mathbf{S}_j). \tag{11}$$

In a real-time processing system, the change in joint probability in a particular demon through adding each newly sampled state-pair is of particular interest.

Let

$$\Lambda_{i+1}(X) = L(X \mid \mathbf{S}_1 \cap \mathbf{S}_2, \ldots \mathbf{S}_i \cap \mathbf{S}_{i+1}),$$

thus

$$\Lambda_{i+1}(X) = L(X \mid \mathbf{S}_1 \cap \mathbf{S}_2, \ldots \mathbf{S}_i) + L(X \mid \mathbf{S}_{i+1})$$

$$= \Lambda_i(X) + L(X \mid \mathbf{S}_{i+1}). \tag{12}$$

Thus, for each demon at any instant in time, t_j, we need simply retain the summed logit activity up to the previous time frame, $\Lambda_{j-1}(X)$, and compute and add the logit probability of the current state-pair corresponding to that demon, $L(X \mid \mathbf{S}_j)$. It is assumed that this computation occurs in parallel for all demons to be considered as possible responses. Initially there may be more possible demons than can be processed in real-time, and input state-pairs may need to be retained in an input buffer. However, as more information is added, it will become clearer from the set of summed logit activities which demons are highly unlikely and which are highly likely. A processing scheduler or "master demon" may be envisioned which both decides which demons to continue processing and finally which is sufficiently likely relative to all others to be taken as the correct response. As more demons are removed from the possible set, more processing capacity can be allocated to the remainder, and the backlog in the input buffer cleared. Neither parallel processing nor a "master demon" form part of the current implementation, but this point will be returned to in discussing the relationship to current psychological models of speech perception. We may note for now that Marslen-Wilson (pers. comm.) has shown that for a random selection of 80 English words, acoustic information in the first syllable was enough to reduce the possible set of corresponding words from the entire English lexicon to an average size of 30 words.

Some practical considerations

It is only possible to compute a meaningful value for $L(X \mid \mathbf{S}_i)$ if neither w_{xi} nor n_{xi} are zero. Unfortunately this may not be the case. Often a state-pair vector in an unknown stimulus will not have appeared in the training corpus and $P(X \mid \mathbf{S}_i)$ will be undefined. In other cases either w_{xi} or n_{xi} will be zero, and $P(X \mid \mathbf{S}_i)$ thus zero or one; no meaningful value can then be assigned to $L(X \mid \mathbf{S}_i)$, and, in physical terms, allowing such extreme values for $P(X \mid \mathbf{S}_i)$ would mean that we consider one single state-pair as sufficient evidence for the recognition or rejection of a particular demon.

It would be desirable to make the system more flexible and less sensitive to the influence of such short duration information. Given any $\mathbf{S}_i$ we therefore need to search for the nearest point in state-pair vector space giving information about w_{xi} and n_{xi}. With the very simple parameter space used, a correspondingly simple search strategy was chosen. Firstly, rather than seeking the *nearest* point to $\mathbf{S}_i$, the next stored point above $\mathbf{S}_i$ is taken if no exact match is found (the one-dimensional state-pair vector being considered as an unsigned integer). Secondly, inspecting a section of the data built up for the "one" demon given in Table III(a), we see that there tend to be sequences of points along our single dimension where either w_{xi} or n_{xi} are zero. Since in neither case can we compute a meaningful value of $L(X \mid \mathbf{S}_i)$, these runs were added together and assigned to the numerically highest $\mathbf{S}_i$ [see Table III(b)]. Given the rule that search is to the nearest point above any $\mathbf{S}_i$, all $\mathbf{S}_i$ mapping onto any of the points in this run will now correspond to this new point. This "collapsing" of the stored state-pair vector stack must of course be done separately for each demon, the runs of zeros in each column being different for different demons.

Table III (a) A section of the state-pair vector stack in the "one" demon. (b) Collapsed vector state for the "one" demon

	(a) "One" demon		(b) Collapsed demon	
State-pair vector	Words	Non-words	Words	Non-words
$\mathbf{S}_i$	w_{xi}	n_{xi}	w_{xi}	n_{xi}
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
01110101110001010001	9	0		
01110110010000000001	2	0		
01110110010001000001	3	0		
01110110010001010001	6	0	20	0
01110110010001010001	1	1	1	1
01110110110000010101	0	1		
01110111010000010101	0	3	0	4
01110111110000010001	4	0	4	0
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.

Table IV gives the number of state-pairs remaining in the collapsed vector stack for each demon. This still does not solve the problem of zero values of w_{xi} or n_{xi} which still remain, so the same "search above" rule was extended, and during recognition search continues, adding w_{xi} and n_{xi} from stored state-pair vector information corresponding to and immediately above $\mathbf{S}_i$ until both are non-zero. A consequence of the collapsing of the vector stacks described above is that no more than two consecutive stored state-pair vectors need be combined to be able to derive a meaningful value for $L(X \mid \mathbf{S}_i)$.

Table IV State-pairs in each demon after collapse

Demon	1	2	3	4	5	6	7	8	9
Word tokens	12	8	11	11	13	13	15	11	11
Word frames	434	304	436	383	514	622	671	340	663
Non-word tokens	93	97	94	94	92	92	90	94	94
Non-word frames	3933	4063	3931	3984	3853	3745	3696	4027	3704
State-pairs	308	195	306	215	232	280	420	221	319

Finally some mention needs to be made of the term $P(X)$ in (9). With a limited equiprobable vocabulary this will be a constant, and it was chosen to omit this from the computation of $\Lambda_i(X)$ in this implementation. This corresponds to the assumption that $L(X)$ is zero, and thus that $P(X) = 0.5$. A constant per unit time may be subtracted from the $\Lambda_i(X)$ so obtained to transform the performance of these refreshingly optimistic demons to true cumulative logit probability. However, this in no way affects comparison of the *relative* level of activity of the demons. Indeed since the demons are independent and have no knowledge of one another's existence, an assumption of $P(X) = 0.5$ seems not unreasonable, corresponding well with their experience of the world being limited to the two categories "word" and "non-word".

ERIS – initial results

Figure 6 shows the basic form of ERIS computer output. For simplicity, only the performance of the "one" demon on the test token of "one" is illustrated. The lower part of the figure shows $L(\mathbf{S}_i \mid X)$ for each 10 ms input frame, positive values indicating the extent to which that state-vector pair favours this token being an example of "one", and negative values indicating the extent of mismatch. It can be seen that in this case almost all state-vector pairs in the stimulus make a positive contribution. The upper part of the figure shows $\Lambda_i(X)$, the cumulative total of $L(\mathbf{S}_i \mid X)$. The rapid rise of $\Lambda_i(X)$ contrasts sharply with Fig. 7 which indicates the corresponding performance of the "two" demon on this same token of "one". The cumulative logit probability rapidly becomes so unfavourable that the values can no longer be plotted on the axes used.

Figure 8(a) combines the performance of all nine demons on this same test token of "one", and Fig. 8(b)–8(i) for test tokens of "two" to "nine" respectively. Only the cumulative plots are given, and for clarity the "correct" demon, that is the one trained to recognise digits corresponding in type to the test token, is drawn with a dotted line. The competing activity of the demons can be clearly seen as time proceeds. A first point to note in these original unselected displays is that every token is successfully recognised, in the objective sense that the final value of $\Lambda_i(X)$ is highest for the nominally correct demon, and in the subjective sense that performance looks convincingly better for that demon.

A number of further characteristics of the results are of considerable interest. Firstly, within 150 ms of the onset of a test stimulus, most of the demons have become so unlikely as possible candidates that their $\Lambda_i(X)$ can no longer be plotted on the axes used. In addition to the nominally correct demon there remain at most one or two which actually

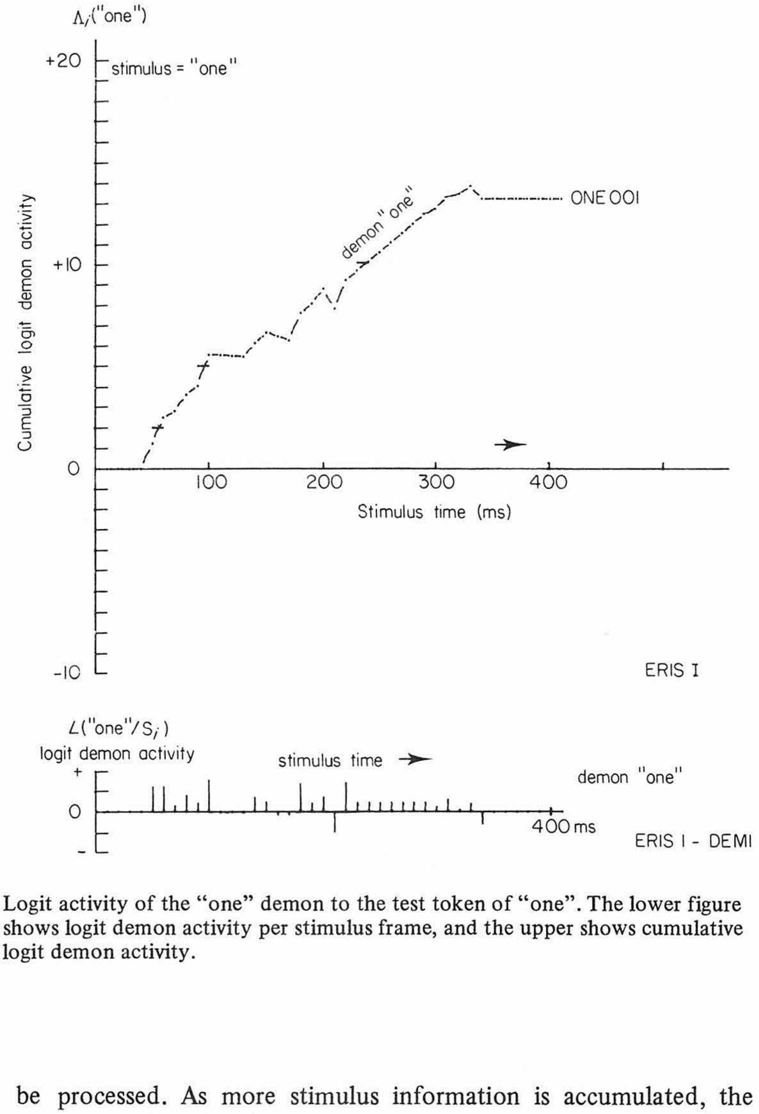

Figure 6 Logit activity of the "one" demon to the test token of "one". The lower figure shows logit demon activity per stimulus frame, and the upper shows cumulative logit demon activity.

need to continue to be processed. As more stimulus information is accumulated, the evidence in favour of the correct demon differentiates it more and more from the others, and a decision can be made well before the end of the target. In some cases such an early decision may not only be possible, but preferable. In the current corpus, the test token of "five" contains some low energy information at the end which corresponds better to the "one" demon, and although in this case it is probably a consequence of the crude parameter quantisation and representation used, we have already seen that in human speech recognition such early decisions may also occur. If we consider, for example, recognition of the names of days of the week, it is clear that an optimal recogniser will attach more significance to matches early in the word and less to the non-discriminative "-day" suffix. Rather than proposing the existence of complex optimisation routines, it is attractive to suggest that the left-to-right structure of the speech signal, the real-time nature of the recognition process and the contents of the lexicon themselves supply this optimisation.

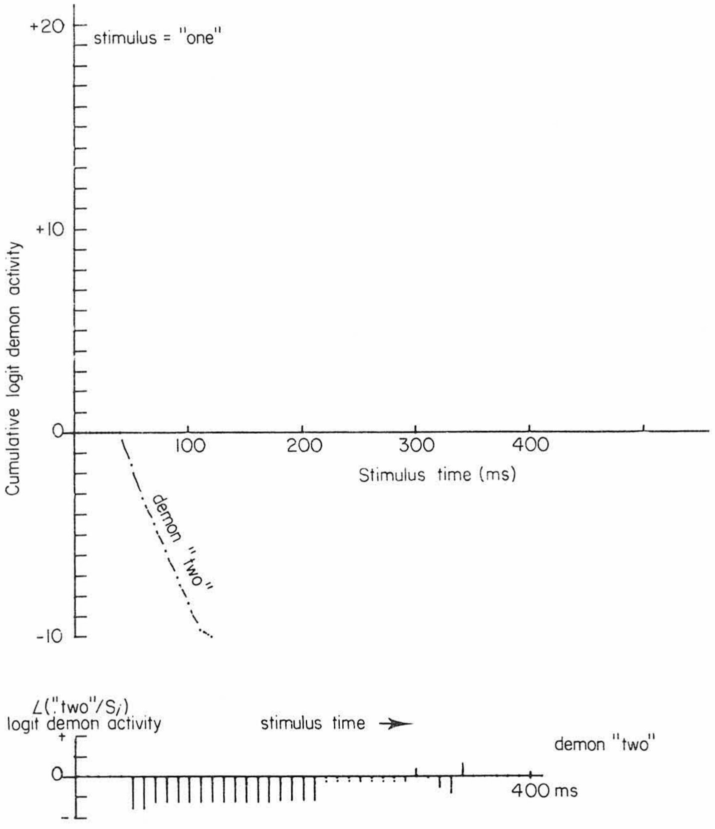

Figure 7 Activity of the "two" demon in response to the test token of "one".

Discussion

Considering its simplicity, the success of the ERIS implementation shows great promise for the applicability of a context-sensitive code in speech recognition. The coding seems ideally suited to real-time processing, and to exhibit many of the characteristics found in human speech recognition, including the ability to make a response decision before the end of the acoustic stimulus. A number of problems which are not of intrinsic concern to context-sensitive coding have not been tackled. These include such points as an optimised choice of stimulus parameter space, a perceptually meaningful distance measure for mapping between points in this space, and speaker normalisation. These must ultimately form a component of *any* model attempting to emulate human speech recognition, but various other problems arise from discarding an absolute time dimension, and these are of immediate concern.

Firstly, the system is not capable of distinguishing stimuli varying only in the duration of a steady-state segment. For example, in Dutch, the difference between the words "tak" (branch) and "taak" (task) may be cued solely by changing the duration of the vowel /a/ without any changes in spectral quality (Nooteboom & Doodeman, 1980). For simplicity, let us assume that the /a/ is totally steady-state, of 60 ms duration in the short "a" and 110 ms in the long "aa". Since there will be five steady-state /a/–/a/ state-pairs in each token of "tak" and 10 in each of "taak", *each* occurrence of an /a/–/a/ state-pair will favour the presence of the long vowel over the short by a factor of 2 to 1. This will be so even when the state-pair originated

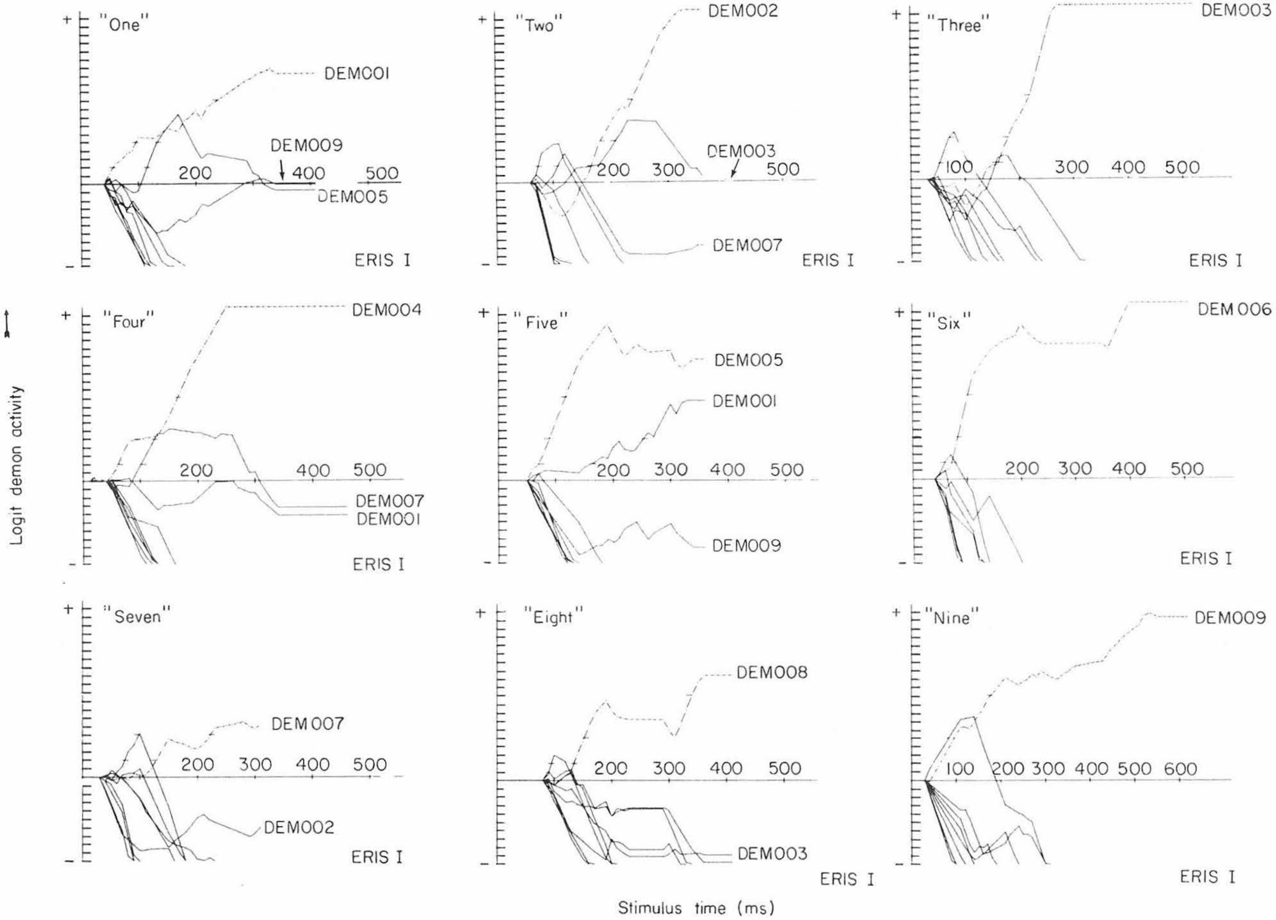

Figure 8 Response of all nine demons to a test token of each of the digits "one" to "nine" in turn. In each tableau the demon corresponding in type to the test token has been drawn dotted.

from a token of "tak". A similar problem is found not only with steady-state information, but in distinguishing stimuli with repeated segments. In an exactly analogous way, it will prove difficult to set up independent demons for "da" and "dada". Although we could design a "long vowel detector" which adds an extra stimulus feature when vowels exceed a particular length, this *post hoc* solution cannot deal with repeated segments, which would require some independent stimulus length normalisation procedure. Before considering such separate solutions to these problems, let us turn to the second shortcoming of ERIS and see if a joint solution can be found.

Context-sensitive coding, by removing the necessity of determining or allocating word onsets or offsets in the unknown stimulus, gives a recognition system a very attractive property. A segment late in a word which corresponds to the onset of another word can also potentially activate the demon corresponding to the second word. For example, the *final* /n/ in a token of "one" may excite the state-pairs corresponding to the *initial* /n/ in the "nine" demon. Each demon operates independently, and final decisions about the presence or absence of a word are made by a supervising "master demon". In this case, the level of activity in the "one" demon, high even before the occurrence of /n/, would result in the recognition of "one" and the consequent rejection of "nine" as a possible response. However, if "one" were not a lexical entry in the language, there would be no corresponding demon, and the /n/ might well be from the onset of "nine". A context-sensitive code can thus admirably deal with initial hesitations, "um"s and "err"s. However, this temporal flexibility also operates in the reverse direction, and here we encounter a second problem. Not only will the final segment of a stimulus excite demons having that as the initial segment in their corresponding words, but the *initial* segment of a stimulus will excite demons having that segment in *final* position in their words. For example, the "one" demon will be excited by the initial /n/ in "nought". Although some temporal flexibility in segment location may be desirable or even essential, this problem will clearly become quite severe with longer, polysyllabic words.

These problems may be summarised as "how can we deal with a dimension analogous to stimulus time without having to return to the use of stimulus time itself, with its associated problems?". Figure 9 illustrates the idealised performance of the "correct" demon on its corresponding stimulus, and something at least approximating this response can be observed in Fig. 8. The proposed solution is therefore to use the activity of each demon itself as its own *input* parameter, analogous to stimulus time; in effect a "bootstrap" parameter. In contrast with the other input parameters, this will have a different value for each demon, and thus, also in contrast with any measure of stimulus time, one demon may consider a stimulus segment as corresponding to a state-pair vector early in its associated word, while at the same time another treats the same segment as corresponding to a final part of its word. Segments later in a word should only be encountered in training with high demon activity, and thus the associated demon will only respond to these segments when it has already been activated (by earlier segments). Conversely, the demon will become less and less sensitive to earlier segments in its associated word as its activity rises. Thus, following /ta/, the "tak" demon will be optimally sensitive to /k/ and the "taak" demon to /a/. A demon for the word "England" will not initially be excited by "and", but following "Engl-" it will be optimally sensitive to "-and" and not to a repeat of "Engl-".

Just as with other acoustic parameters, some flexibility will be needed, with an appropriate distance measure also operating along this "demon activity" dimension. It is not proposed that a demon should initially be totally insensitive to later segments, or totally reject repeated earlier segments. However, an *optimal* response should be produced by the optimally correct stimulus.

ERIS II – the demon activity dimension

This "bootstrap" dimension has been incorporated into a pilot version of ERIS, and like all other parameters used, this demon activity dimension was quantised, in this case into five equal steps as indicated by the five regions shown in Fig. 9. As each of these activity boundaries is crossed, the demon switches over to a new set of state-pair vector frequencies, and the other parameters are thus nested under this new dimension. Each set of vector frequencies will be termed a *sub-demon*. Some economies in the training phase meant that each sub-demon contained both information specific to that level of demon activity and also undifferentiated state-pair vector frequency information from the original demon. Figure 10 shows provisional results, comparing the performance of the original "seven" demon and the five new sub-demons on the test-token of "seven". It can be seen that compared with the original demon, each of the new sub-demons is more optimally sensitive to an appropriate beginning, middle or end segment of the test token. However, the cumulative logit activity plot shows that this analogue of stimulus time requires an analogue of time normalisation. Due to the lack of a sufficiently good match on this particular stimulus token, the cumulative activity of the sub-demons on the test token of "seven" never rises high enough to "move up" higher than the first sub-demon (level I). Since, as in this case, the earlier sub-demons which remain in operation may actually find later segments to be negative evidence for the occurrence of their associated word, the end result may be poorer than with the undifferentiated demon. Rather than indicating the failure of this approach, this suggests that rather more sophistication needs to be incorporated in the use of the demon activity dimension. One possibility is that relative rather than absolute activity level should be used to decide on the appropriate sub-demon to use, feedback of a supervising "master demon" giving information on the extent of match of the text token relative to all other demons.

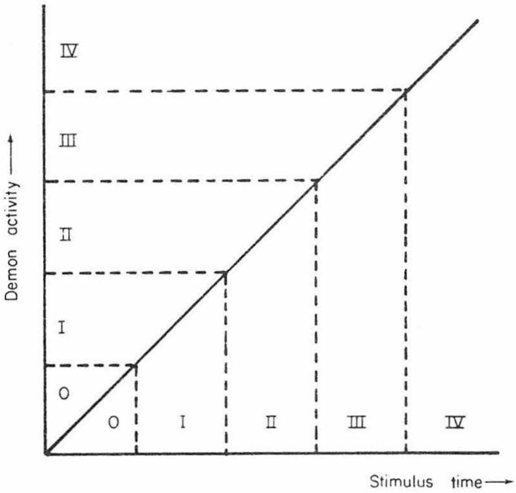

Figure 9 Demon activity plotted against stimulus time for the "correct" demon. For an explanation of regions 0 to IV, see text.

Although demon activity levels were quantised for purely practical reasons, the resulting sub-demons now constitute an intermediate processing level between the acoustic input

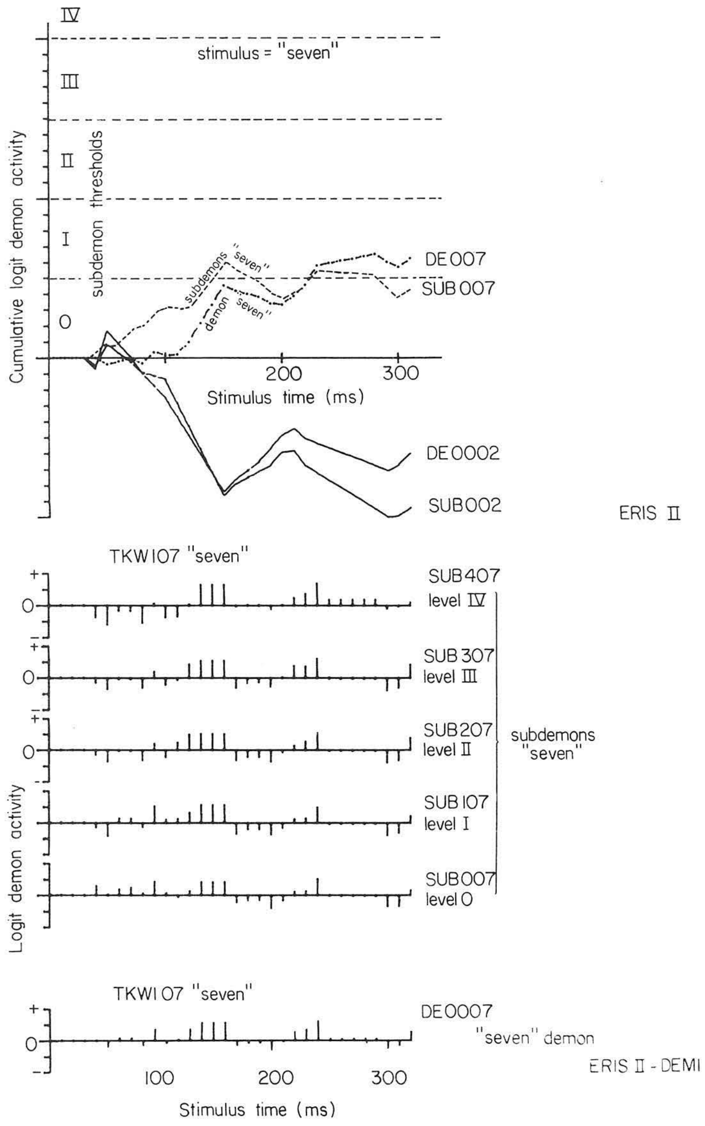

Figure 10 Response of "seven" demon and sub-demons to the test token of "seven".

and word recognition. There are a number of reasons why it would be desirable or useful to incorporate some such intermediate level in the recognition process, but these are often confused with a particular theoretical standpoint. It would seem more elegant to make use of the common features of the language and store words in a more compact form than independent sets of acoustic attributes, but arguments simply based on economy of storage should be regarded with some suspicion. It may also be plausible to suggest that phonological rules could be a consequence of the properties of such a storage and access system. A second and more important reason for requiring some intermediate level is the simple observation that we can learn new words and perceive and repeat nonsense words.

As experienced language users, we can often learn a new word on a single repetition, and this would be hard to explain if learning involved setting up a totally new acoustic representation. If such were the case, we should expect that new words would fall on deaf ears until information had been assembled from a number of tokens of this word. Instead, we find that although a listener can and does know whether a word forms a part of his lexicon, he can also repeat and subsequently recognise both unknown words and nonsense words. Such recognition becomes progressively more difficult as these strings differ more and more from the forms of his own language, as students of any foreign language will be aware.

There are thus reasonable motives for supposing *some* intermediate level between the acoustic input and word or morpheme recognition, but little or none for supposing this to correspond to phonemes. Additionally, although the phenomena described above demonstrate that it is possible for speech perception to function in a manner which suggests or requires sub-lexical units of analysis, they do not show any such units to be involved in the rapid, "hyperaccurate", recognition of known words in continuous speech. In fact, the observation that we can recognise a mispronounced word for what it should have been while knowing it to have been mispronounced demands the relative independence of lexical access and some level of intermediate representation.

The size and nature of such intermediate units remains to be determined. Attempts at establishing a phonemic representation for an unknown acoustic input, or for synthesising speech from such a representation, result in a multiplicity of special rules and exceptions, multiple representations and considerable ambiguity and confusion. It seems likely that the smallest useful representation will be one which maintains a reasonably invariant relation to the acoustic waveform, and as such, "diphones" or Wickelgren's (1969) "context sensitive allophones" would appear to be possible candidates. A similar point has been made by Klatt (1979).

It is suggested that the most fruitful experimental and theoretical approach is one which does not make any *a priori* assumptions on the size and nature of intermediate units, but instead examines the recognition process as an acoustic-lexical mapping. We may subsequently ask what intermediate levels might be reasonable, useful or necessary in this process. ERIS demonstrates the viability of such an approach, and it remains to be seen in precisely what way demons can and should be represented by common sets of state-pair vectors.

Context-sensitive coding and models of human speech recognition

Little attempt has been made to conceal the fact that ERIS and her demons constitute an attempt at implementation of a subpart of Morton's (1964, 1969) Logogen Model. Since we are specifically concerned with the early acoustic analysis component of the model, we come up against an incompleteness in the original specification of the Logogen Model, namely that the nature of *acoustic attributes* is left unspecified. This present paper has used a computer simulation to test the viability of a simple form of context-sensitive coding as a representation of the acoustic signal for speech recognition. By considering various other aspects of human speech recognition, principally those revealed by the work of Marslen-Wilson and his associates, various extensions have been made to or proposed for this simulation, resulting in it differing from the Logogen Model not just in terms of completeness, but also in more fundamental principles. The most important of these are the inclusion of negative stimulus information, and the proposed external availability of demon activity levels to some "master demon" or processing scheduler. It has not however been the intention of this paper to develop a complete model of word recognition, and the precise ways in which it suggests changes or extensions to the Logogen or any other model of speech perception remain to be specified.

It needs perhaps to be emphasised that the essence of this paper is the empirical success in using a context-sensitive code in speech perception. The choice of the Logogen framework as a means of testing such a code in no way detracts from its general applicability in speech recognition. Those working with highly structured state-transition networks may see the first-order context-sensitive code as a vastly simplified reduction of such data bases. The results presented here then naturally question how much of the complexity of such sophisticated structures and their associated search algorithms are necessary. It will be interesting and exciting to see how successfully the impossibly simple idea presented here may be applied to such systems.

This paper also clearly illustrates the value of computer simulations such as ERIS as tools in illuminating the processes required in psychological models of speech recognition. Since ERIS was not implemented with the intention of building a complete word recognition system, nor meant to be a direct copy of any psychological model (though its close affinity to the Logogen Model is of course acknowledged), it provides an interface between theory and real speech. A number of problems which can remain unspecified in a psychological model must be faced; even if we choose to deal with them at present in an *ad hoc* fashion, we are at least forced to recognise what is lacking in our current knowledge and what needs to be properly treated in the future. Similarly, a number of practical programming and hardware considerations which would have to be dealt with in a working word recognition system need not be optimally solved. The computer simulation becomes a dynamic extension to the functional diagram, giving insight into how our ideas operate in their model world.

I am indebted to John Morton whose imagination, energy and enthusiasm led to many of the ideas worked out and presented in this paper. I am also grateful to friends and colleagues at IPO and elsewhere, and in particular to Louis Goldstein, William Marslen-Wilson and Sieb Nooteboom, for stimulating and valuable discussion, whether in the form of enthusiastic encouragement or wise words of caution. Finally, thanks are due to the aged P9202 (DDP-516) computer for service above and beyond what could reasonably be expected from an old lady of her generation.

References

Bakis, R. (1974). Continuous-speech word spotting via centisecond acoustic states. *IBM speech processing group, report RC 4788.*

Browman, C. P. (1978). Tip of the tongue and slip of the ear: implications for language processing. *UCLA Working Papers in Phonetics*, **42.** Los Angeles: University of California.

Fant, C. G. M. (1969). Stops in CV syllables. *STL QSPR 4/1969*, 1–24. Stockholm: Speech Transmission Laboratory.

Forney, G. D., Jr. (1973). The Viterbi Algorithm. *Proceedings of the IEEE*, **61**, 268–278.

Klatt, D. H. (1979). Speech Perception: a model of acoustic–phonetic analysis and lexical access. *Journal of Phonetics*, **7**, 279–312.

Liberman, A. M., Cooper, F. S., Shankweiler, D. P. & Studdert-Kennedy, M. (1967). Perception of the speech code. *Psychological Review*, **74**, 431–461.

Lieberman, P. (1963). Some effects of semantic and grammatical context on the production and perception of speech. *Language and Speech*, **6**, 172–187.

Lindblom, B. E. F. (1963). Spectrographic study of vowel reduction. *Journal of the Acoustic Society of America*, **35**, 1773–1781.

Lowerre, B. T. (1976). *The HARPY speech recognition system.* Unpublished PhD thesis, Carnegie-Mellon University.

Marslen-Wilson, W. (1973). Speech shadowing and sentence perception. Unpublished PhD thesis, Massachusetts Institute of Technology.

Marslen-Wilson, W. & Welsh, A. (1978). Processing interactions and lexical access during word recognition in continuous speech. *Cognition*, **10**, 29–63.

Miller, G. A., Heise, G. A. & Lichten, W. (1951). The intelligibility of speech as a function of the context of the test materials. *Journal of Experimental Psychology*, **41**, 329–335.

Morton, J. (1964). A preliminary functional model for language behaviour. *International Audiology,* **3**, 216–225. (Reprinted in *Language*, R. C. Oldfield & J. C. Marshall, (Eds.). London: Penguin Books, 1968).

Morton, J. (1969). Interaction of information in word recognition. *Psychological Review,* **76**, 165–178.

Morton, J. & Broadbent, D. E. (1967). Passive versus active recognition models or is your Homunculus really necessary? In: *Models for the Perception of Speech and Visual Form,* Wathen-Dunn, W. (Ed.) pp. 103–110. Cambridge, Mass.: MIT.

Morton, J. & Long, J. (1976). Effect of word transitional probability on phoneme identification. *Journal of Verbal Learning and Verbal Behaviour,* **15**, 43–51.

Neisser, U. (1967). *Cognitive Psychology.* New York: Appleton-Century-Crofts.

Nooteboom, S. G. & Doodeman, G. J. N. (1980). Production and perception of vowel length in spoken sentences, *Journal of the Acoustic Society of America,* **67**, 276–287.

Reddy, D. R. (1976). Speech recognition by machine: a review. *Proceedings of the IEEE,* **64**, 501–531.

Rubin, P., Turvey, M. T. & van Gelder, P. (1976). Initial phonemes are detected faster in spoken words than in spoken nonwords. *Perception and Psychophysics,* **19**, 394–398.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal,* **27**, 379–423, 623–656.

Streeter, L. A. & Nigro, G. N. (1979). The role of medial consonant transitions in word perception. *Journal of the Acoustic Society of America,* **65**, 1533–1541.

Swadesh, M. (1934). The Phonemic principle. *Language,* **10**, 117–129.

Vogten, L. L. M. & Willems, L. F. (1977). The formator: a speech analysis-synthesis system based on formant extraction from linear prediction coefficents. *IPO Annual Progress Report,* **12**, 47–62. Eindhoven, The Netherlands.

Warren, R. M. (1968). The verbal transformation effect and auditory perceptual mechanisms. *Psychological Bulletin,* **70**, 261–270.

Warren, R. M. & Gregory, R. L. (1958). An auditory analogue of the visual reversible figure. *American Journal of Psychology,* **71**, 612–613.

Wickelgren, W. A. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behaviour. *Psychological Review,* **76**, 1–15.

Wickelgren, W. A. (1972). Coding, retrieval, and dynamics of multitrace associative memory. In: *Cognition in Learning and Memory,* Gregg, L. W. (Ed.). New York: Wiley.