Cognition 52 (1994) 189–234

Shortlist: a connectionist model of continuous speech recognition

Dennis Norris

Medical Research Council Applied Psychology Unit, 15 Chaucer Road, Cambridge CB2 2EF, UK.

Received July 28, 1993 final version accepted February 2, 1994

Abstract

Previous work has shown how a back-propagation network with recurrent connections can successfully model many aspects of human spoken word recognition (Norris, 1988, 1990, 1992, 1993). However, such networks are unable to revise their decisions in the light of subsequent context. TRACE (McClelland & Elman, 1986), on the other hand, manages to deal appropriately with following context, but only by using a highly implausible architecture that fails to account for some important experimental results. A new model is presented which displays the more desirable properties of each of these models. In contrast to TRACE the new model is entirely bottom-up and can readily perform simulations with vocabularies of tens of thousands of words.

1. Introduction

In contrast to written language, speech is an inherently temporal signal. In the case of written language all of the letters in a word are available for processing simultaneously. In speech, the information in a word must necessarily arrive sequentially. The location of the boundaries of written words is greatly facilitated by the presence of white spaces between words. In speech there tend to be few reliable cues to word boundaries. But, despite the very different nature of the problems involved in recognising written and spoken language, most psychological theories of word recognition have tended to adopt the convenient fiction that

Part of this work was carried out while the author was visiting the Max Planck Institute for Psycholinguistics, Nijmegen. This work was supported in part by grant E304/148 from the Joint Councils Initiative. Thanks to Uli Frauenfelder, Anne Cutler and James McQueen for valuable discussions.

SSDI 0010-0277(94)00622-R

speech is just written language with phonemes instead of letters. However, in the last fifteen years models have emerged which have begun to pay attention to the distinctive character of spoken language. In particular, Marlsen–Wilson and his colleagues (Marlsen-Wilson, 1987; Marlsen-Wilson & Welsh, 1978) have developed the Cohort model with its emphasis on the "left-to-right" nature of speech recognition and on how the process of recognition unfolds over time. In the TRACE model McClelland and Elman (Elman & McClelland, 1986; McClelland & Elman, 1986) have extended this concern with the temporal dynamics of spoken word recognition. Additionally, TRACE provides a solution to the problem of segmenting the continuous speech stream into words. Indeed, TRACE is sometimes thought of as a computational implementation of some of the ideas first expressed in the Cohort model.

In recent years TRACE has become the most widely applied model of human spoken word recognition. The success of TRACE as a psychological model is probably attributable to two main factors. First, TRACE is very broad in its coverage. It successfully simulates a broad spectrum of psychological data ranging from compensation for coarticulation to data on word recognition points. Second, TRACE is computationally explicit. There is no room for debate as to the predictions TRACE makes. The code for TRACE has been widely distributed and other researchers (e.g., Frauenfelder & Peeters, 1990) have been able to make extensive use of TRACE simulations in their own work.

However, despite its success, TRACE has not gone unchallenged. Some of the central theoretical assumptions of TRACE have aroused considerable controversy. TRACE is an expression of a highly interactive view of spoken word recognition in which there is a continuous two-way flow of information between lexical and phonemic processing. This interactionist view has received a strong challenge from bottom-up theories in which the processes involved in phoneme recognition are completely autonomous and receive no top-down feedback from lexical analysis (Cutler, Mehler, Norris, & Segui, 1987; Massaro, 1989). In the last few years a number of studies have produced results which favour the autonomous view over the interactionist standpoint represented by TRACE (Burton, Baum, & Blumstein, 1989; Burton, & Blumstein, MS; Cutler et al., 1987; Frauenfelder, Segui, & Dijkstra, 1990; McQueen, 1991a).

An additional problem for TRACE is that it employs an architecture of rather questionable plausibility. In TRACE the problem of time-invariant recognition is solved by duplicating the entire lexical network many times. A theory which could avoid the need to duplicate lexical networks would represent a considerable advance over TRACE.

The present paper develops a new model of spoken word recognition which addresses these two central deficiencies of TRACE. Consistent with the empirical data, the model is entirely bottom-up in its operation. In many respects the model can be considered to be an implementation of the bottom-up race model of Cutler

and Norris (1979). The model also addresses the problem of the plausibility of the TRACE architecture. Like TRACE, the Shortlist model relies on competition between lexical candidates tied to specific locations in the input. However, in Shortlist the competition takes place within a small, dynamically generated network which only ever considers a handful of lexical candidates at any one time. The structure of the model enables it to perform simulations using realistically sized vocabularies. Simulations are presented which show that the model performs well with large vocabularies even when the input is degraded or potentially ambiguous. A bottom-up architecture is no barrier to efficient performance. In word recognition, top-down feedback from the word to the phoneme level is redundant because all of the crucial lexical constraints can operate entirely within the lexical level itself.

2. The data

Cutler et al. (1987) reported two findings which require revisions of TRACE. First, they found that phoneme monitoring latencies to word-initial phonemes were faster than to phonemes beginning non-words. McClelland and Elman (1986) argued that effects of lexical status should not manifest themselves on word initial phonemes because the lexical activation will not have had time to build up sufficiently to feed back down to the phoneme level. Second, Cutler et al. showed that the effect of lexical status was dependent on the composition of the stimuli. Effects of lexical status emerged only in lists where the items varied in number of syllables. In lists of monosyllables the effects disappeared. Cutler et al. interpreted their results in terms of a race model (Cutler & Norris, 1979) in which attention could be shifted from a phonemic to a lexical analysis. They suggested that the monotony of the monosyllabic lists led subjects to attend primarily to a phonemic analysis of the input, whereas in the more varied lists they attended more to the results of a lexical analysis. An attentional explanation of this kind fits in well with a race model where there are two sources of information about phoneme identity. Attention can be shifted between the phonemic level and the lexical level. However, in TRACE there is only a single source of phoneme identity information. Phonemes can only be recognised by reading out information from the phoneme nodes. To accommodate these results TRACE would need to be modified so that all of the top-down word-phoneme connections could be altered to produce more top-down activation in the more varied lists. Such a move would account for the data but would be harder to motivate than the attentional explanation offered by Cutler et al.

Frauenfelder, Segui, and Dijkstra (1990) also used the phoneme monitoring task to examine the predictions of TRACE. They measured monitoring latencies to phonemes occurring after the uniqueness point of a word. In some instances

the target phoneme was altered to form a nonword; for example, the /l/ in *vocabulaire* was altered to a /t/ forming *vocabutaire*. Reaction times to the /t/ in *vocabutaire* were compared to reaction times to the /t/ in the control nonword *socabutaire*. According to TRACE, top-down feedback from the lexical node corresponding to *vocabulaire* should inhibit identification of the /t/ in *vocabutaire* but not in *socabutaire* where there should be only minimal lexical activation. However, although Frauenfelder et al. found facilitatory lexical effects in the word conditions in their study, they found no evidence of the predicted inhibition. This absence of inhibition is, however, exactly what is predicted by an autonomous theory such as the race model of Cutler and Norris (1979) in which there is no top-down influence of lexical information on phoneme identification. According to Cutler and Norris, phoneme identification is a race between a phonemic route and a lexical route in which the phonological representation of a word is accessed from the lexicon. The lexical and phonemic routes are completely independent and responses are determined by a first-past-the-post race. If the phonemic route wins the race, then lexical information will have no influence on the outcome. So, identification of the /l/ in *vocabulaire* will be faster than in the non-word *socabulaire* because words benefit from the operation of the faster lexical route. However, the /t/s in *socabutaire* and *vocabutaire* will be identified equally quickly because both will be identified by means of the phonemic route.

Concerns over the importance of top-down feedback have also been raised by recent studies by Burton et al. (1989), Burton and Blumstein (MS), and by McQueen (1991a). These studies suggest that top-down effects of lexical information on phoneme identification may be far less pervasive than a highly interactive model like TRACE would suggest. Top-down effects may well be dependent on the quality of the stimulus and may only emerge when the stimulus is degraded in some way, either by low pass filtering or by the removal of phonetic cues. Even then, the effects do not appear to be consistent (for a review of lexical effects on phonetic categorisation see Pitt & Samuel, 1993). The study by McQueen investigated the effects of lexical information on the categorisation of word final fricatives. According to TRACE, the top-down effects of lexical activation on phoneme perception should be at their strongest in word-final position. Subjects in McQueen's study heard stimuli in which the final fricative varied on a continuum between /s/ and /ʃ/. TRACE predicts that subjects hearing stimuli on a /fɪs/ – /fɪʃ/ continuum should show a shift in their categorisation function such that ambiguous stimuli are more likely to be identified as /ʃ/. The top-down activation from *fish* should bias the perception of the ambiguous phoneme. This bias should be present even for stimuli presented under good listening conditions. However, the predicted lexical bias was only present when the stimuli were low-pass filtered at 3000 Hz. Furthermore, the lexical bias was most apparent in the case of the fastest responses. McQueen argues that this pattern of results is contrary to the predictions of TRACE but in

line with the race model of Cutler and Norris. In TRACE, lexical bias should be dependent on the activation of the lexical node. Lexical activation should grow over time. Therefore later responses should show a greater lexical bias because there will be more top-down feedback. According to the race model, lexical effects will be most apparent where the lexical route tends to be faster than the phonological route. Lexical effects should therefore be largest in the fastest responses.

The common thrust of the empirical evidence against TRACE is a concern that McClelland and Elman may have placed too much emphasis on the importance of top-down information. Certainly, lexical information may influence phoneme monitoring and categorisation responses under some circumstances but there is very little evidence to suggest that this is mediated by an interaction between lexical and phonemic information of the form incorporated in TRACE. The strongest support for the interactive view comes from a study of compensation for coarticulation by Elman and McClelland (1988). Elman and McClelland pointed out that, within TRACE, activation of a phoneme node caused by top-down information will be indistinguishable from activation caused by bottom-up perceptual information. In compensation for coarticulation (Mann & Repp, 1981; Repp & Mann, 1981), the interpretation of one phoneme is biased by the nature of the preceding phoneme. There is universal agreement that this phenomenon must operate at the phoneme level and not the lexical level. So, if the interpretation of the preceding phoneme itself could be influenced by top-down evidence, TRACE has to predict that the preceding phoneme would behave exactly as if it had been activated by perceptual evidence. Therefore, there should still be compensation for co-articulation regardless of whether the evidence for the phoneme is bottom-up or top-down. This is what Elman and McClelland found. However, according to a bottom-up model, lexical information could not possibly feed back down to the phoneme level. A lexically induced bias should never be able to alter the low-level interpretation of a phoneme so as to influence the compensation for coarticulation effect.

However, as McQueen (1991b) has shown, the lexical effects in this study are critically dependent on using slightly degraded stimuli. According to TRACE such effects should be present even with undegraded input. Also, Norris (1992, 1993) has successfully simulated Elman and McClelland's results using a back-propagation network in which there are no top-down connections at all. Indeed, in the network used in one of the simulations presented by Norris there are not even any word nodes. So, there is still little evidence in favour of the kind of top-down interaction embodied in TRACE.

The alternative view, exemplified by the race model of Cutler and Norris, is that behaviour which appears to be interactive is due to the fact that phonemic information can be derived from two sources. Phonemes can be identified either by a direct phonological analysis of the input, or by accessing the word's

phonological representation in the lexicon. In TRACE, of course, there is only one source of phonological information, the activation of the phoneme nodes themselves. There are no lexical entries containing phonological representations, so lexical effects on phoneme identification can only be explained by top-down interaction.

It is worth emphasizing at this point that, despite the substantial differences in the theoretical claims underlying TRACE and the race model, the two theories have proved very difficult to tease apart. By and large, both theories can account for the same set of phenomena. For example, in the Ganong effect (Ganong, 1980) which forms the basis of the studies by Elman and McClelland and by McQueen, the interpretation of an ambiguous phoneme in a string which is ambiguous between a word and a non-word is biased so as to make subjects more likely to identify the phoneme so as to form a word than a non-word. In Ganong's original experiment subjects heard sequences beginning with a phoneme on a continuum between /t/ and /d/. One end of the continuum was a word, the other a non-word. Subjects were more likely to identify ambiguous phonemes in the middle of the continuum as being consistent with the word interpretation than the non-word. For example, on hearing the midpoint of the continuum "type" – "dipe" subjects were more likely to identify the ambiguous phoneme as /t/ than /d/. According to TRACE, this result is due to the top-down activation from the partially activated word node altering the activation of the phoneme node for /t/. According to a race view this result is due to subjects reading out phonological information from the lexical representation of "type" when there is inadequate bottom-up evidence to identify the phoneme clearly. So, although the Ganong effect appears to be due to top-down interaction it can equally well be explained in terms of a race between lexical and phonemic processing.

Although the basic effect of lexical information on phonetic categorisation can be explained be either bottom-up or top-down theories, we have seen that TRACE and the race model do make predictions which differ in important respects. The detailed pattern of results observed by Cutler et al., Frauenfelder et al. and McQueen tend to tip the balance in favour of the race model. Currently, the strongest evidence in support of TRACE comes from the study by Elman and McClelland. However, as has already been mentioned, even this result can be simulated using a recurrent network with no top-down lexical feedback.

3. Time-shift invariance

Although the empirical findings clearly pose problems for TRACE, the most worrying aspect of TRACE is the implausibility of its architecture. In order to demonstrate time-shift invariance, that is to be able to recognise words no matter when in time they begin, TRACE has to employ multiple copies of the basic

lexical network. TRACE needs one copy of the network aligned with each point in time where a word might begin. The basic architecture of TRACE was inherited from the interactive activation model of visual word recognition (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982). The original interactive activation model employed position specific letter detectors. That is, there had to be separate letter nodes for an "A" in word initial position and for an "A" in second position. The model could only deal with four-letter words and therefore had four separate sets of position specific letter nodes. If a word was presented to the network misaligned so that its first letter appeared in slot two, it could not possibly be recognised, because a letter in position two is treated as a completely different object from the same letter in position one. In the case of visual word recognition, position-specific letter nodes might be considered to have some degree of plausibility. Written words are usually bounded by white space. So it might be possible to line input words up relative to the space. The first word after the space is position one, the second is position two, and so forth. Preceding the network with a special alignment process would at least allow it to work. However, the plausibility of the model would still be open to question. But the case of speech is rather different from that of visual word recognition. There are not usually any reliable cues to word onsets. Words can begin at almost any point in the input, so it would be impossible to construct a reliable alignment process to line the network up with word onsets.

To overcome this problem TRACE duplicates the basic word recognition network so that there is a complete lexical network starting at each point where a word might begin. If an utterance has 50 phonemes then TRACE would need 50 lexical networks to process it. Word nodes within these networks are then connected via inhibitory links to ensure that only a single word is recognised in any given stretch of the input. Apart from the problem that this brute force solution lacks subtlety and aesthetic appeal, it also faces another difficulty. Simply duplicating lexical networks is not a general solution to the time invariance problem. If we want to build a system that will recognise any word in an utterance 5 seconds long we could build an array of 50 or so lexical networks, one for each phoneme (potential word onset) in the utterance. But there clearly has to be some limit on the number of lexical networks that we can use and this would place a limit on the length of utterance we could listen to.

A slightly better solution might be to connect the networks together in a ring with a length determined by memory span. The input would simply be cycled round successive networks in the ring. So long as activation decayed before each section of the ring had to be reused, such a system would be able to deal with utterances of unlimited length.

However, we are still left with the awkward feature of duplicated lexical networks. Is it possible to achieve the same results as TRACE with only a single network? One way to perform time-invariant recognition is to use back-propaga-

tion networks with time-delayed connections. Norris (1988, 1990, 1993) has shown how a very simple network architecture can perform time-invariant word recognition using a single network. Other architectures with time-delayed and recurrent connections are now in common use as phoneme recognisers (e.g., Robinson & Fallside, 1988; Waibel, Hanazawa, Hinton, Shikano, & Lang, 1988; Watrous, Shastri, & Waibel, 1987) in automatic speech recognition systems. The network used by Norris is shown in Fig. 1. The networks in Fig. 1(a) and (b) are functionally equivalent. However, the representation in Fig. 1(b) emphasises the network's heritage from a network originally proposed for production of sequences by Jordan (1986).[1]

The network has a single set of input nodes corresponding to a featural description of the input phonemes, and a single set of output nodes, one for each word in the network's vocabulary. The input to this network consists of a featural representation of the phonemes. The features of successive phonemes are presented to the same set of input nodes in sequence. Throughout the presenta-

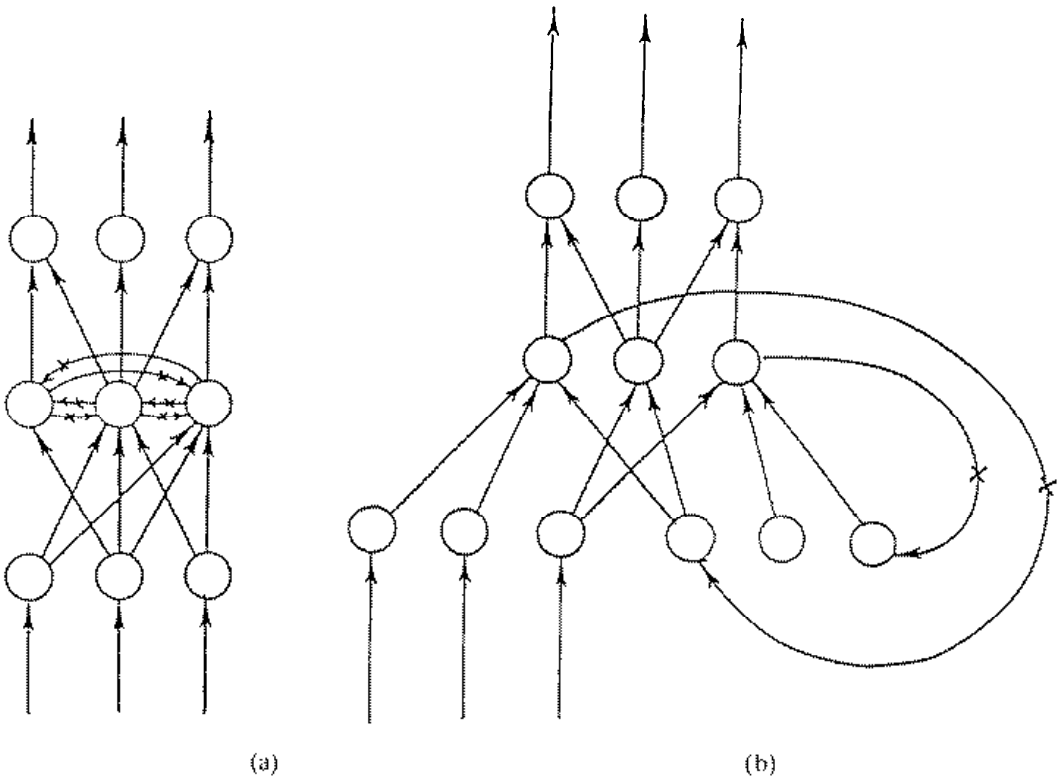


Figure 1. *Two alternative representations of a simple recurrent network. In (a) there are time-delayed weights interconnecting all of the hidden units. In (b) the hidden unit activations are shown as being copied to a set of state units which are, in turn, connected to all hidden units. Links marked "x" have a delay of one time unit (not all connections shown).*

[1] See Norris (1990) for a comparison of the production and recognition architectures.

tion of each word the network is trained to activate a single output node which identifies the word, and all other output nodes are set to zero. The delayed connections in the network ensure that at each point the hidden unit activation generated by the previous input is fed back to the hidden units. So, at all times, the hidden units have access to information about their state on the previous time cycle. The state on the previous cycle was itself determined by the state on the cycle before that. The delayed connections therefore provide the network with a memory for its prior actions and enable it to integrate information across time. Therefore, no matter when in time a word begins, the network will be able to build up the same internal representation of the word and the word will be recognised.

A simple network like this does a remarkably good job at simulating the kind of data that is often cited in support of the Cohort model (Marslen-Wilson, 1980, 1984; Marslen-Wilson & Welsh, 1978; Marslen-Wilson & Zwitserlood, 1989). It will recognise words at the earliest point where they become unique. Before a word becomes unique it will activate all members of the cohort. Words cease to become activated as soon as inconsistent input is received. This kind of architecture is also good at accommodating variations in the rate of input. A network trained to recognise patterns presented at two different rates generalises very well to instances of the same patterns presented at a different rate (Norris, 1990). An interactive activation network has no means of performing such "time-warping" or generalisation across presentation rates.

However, although this network has many desirable properties, it does have one rather serious deficiency which, in fact, is a direct consequence of the decision to use a lexical network with a single set of output nodes. The output of the network at any time effectively represents the network's best bet as to which word is in the input at one particular point in time. Consider what happens if the network receives an input such as *catalog*. When the network processes the /t/ it might activate *cat* while still having little or no activation for *catalog*. By the end of the word *catalog* should be activated and there should no longer be any activation for *cat*. Anyone transcribing the output of the network would therefore simply see two output nodes activated in succession, one corresponding to *cat* and the other to *catalog*.[2] The input *catlog* will also activate two words; *cat* and *log*. Without access to a phonological description of the words there would be no way of knowing that *cat* is just a spurious transient response to the initial phonemes of the word *catalog* in the first case, but a correct identification in the second case. The only way to know that *cat* should be ignored is to be able to examine the phonological representations of both *cat* and *catalog* and to realise that *cat* begins with the same three phonemes as *catalog*. Without access to phonological representations, /kætəlɒg/ and /kætlɒg/ will both be transcribed as containing

[2] *log* will tend to be inhibited by the activation of *catalog*.

two words. Because the network has only a single set of output units it can only indicate its current interpretation of the input. It has no way of going back in time to revise earlier decisions which need to be altered in the light of following context. It cannot represent both the initial response corresponding to the network's best guess based on evidence available at one time, and revised output based on following context.

We could make the network delay its decisions until more context was available, but then it would no longer correctly simulate data showing that words can be recognised almost as soon as they become unique (Marslen-Wilson, 1980, 1984). Note that simply extending the network to have extra outputs representing the network's past history of decisions does not provide a general solution to the problem. It might appear that the network could learn to inhibit *cat* whenever *catalog* is recognised. However, every time a new embedded word such as *cat* is learned the network would have to be retrained with all words in which it is embedded. Also, if *catalog* is misperceived as *cadalog*, *catalog* may be successfully recognised, but the network would also recognise *cad* as it would not have been trained on the relation between *cad* and *catalog*.

A general solution to this problem of how to deal with "right-context" requires that evidence from a particular part of the input can only be used to support a single word. If /kæt/ is being taken as evidence for *catalog* then it can not also be taken as evidence for *cat*. Alternative lexical hypotheses like *cat* and *catalog* need to compete with each other for the available perceptual evidence to ensure that only the best fitting candidate wins through. This is exactly what the lexical level of TRACE does. Word nodes in TRACE compete by means of the inhibitory connections within the word level. Words receiving support from the same input phonemes inhibit each other so that the network's final interpretation of the input is unlikely to contain two words that receive input from the same phoneme. However, TRACE effectively considers all words in the lexicon to be active lexical hypotheses all of the time. Every word in the lexicon is in constant competition with every other word in the lexicon. In fact each word is in competition not only with a complete set of all possible words beginning at the same point in the input, but also with nodes for any words beginning at other positions which would share overlapping input phonemes. For example, in a 50 000 word lexicon in which all words were six phonemes in length, TRACE would need a minimum of 550 000 word nodes to process a word in continuous speech.[3] As TRACE requires bidirectional inhibitory links between every pair of

[3] The word node corresponding to the onset of a 6-phoneme word has to be connected to all other words in the same segment, to word nodes in all of the previous 5 segment positions, because all of these words overlap, and to word nodes in the 5 other segment positions in the remainder of the word. This makes a total of $11 * 50\,000 - 1$ other word nodes that a candidate word node in the middle of continuous speech needs to be connected to. Every pair of nodes corresponding to overlapping words then needs to be connected together.

nodes corresponding to overlapping words, the lexical level alone of this network would require 113 749 724 994 connections. This is the main source of TRACE's implausibility. Because the entire lexicon is involved in competition, the entire lexicon has to be duplicated at each time slice.

So, whereas TRACE suffers from the implausibility of having to use multiple lexical networks to solve the time invariance problem, the recurrent network suffers precisely because it does use only a single network with a single set of lexical nodes. Because it uses only one set of output nodes, it is unable to revise prior decisions in the light of new information.

The limitations of the recurrent net demonstrate that the lexical competition process incorporated into TRACE is not simply a move forced on TRACE by the use of multiple position-specific lexical networks. Any spoken word recogniser, even one which uses only a single lexical network, must be able to compare the merits of competing lexical candidates and to take account of the constraints imposed by overlap between alternative candidates. In automatic speech recognition systems this problem is generally solved by algorithms like dynamic programming (Bellman, 1957) and its descendents (e.g., Chien, Lee, & Chen, 1991; Thompson, 1990; Tomita, 1986). In these techniques the task is generally expressed as being one of finding an optimum path through a word lattice. The word lattice encodes the lexical candidates, their start and end points, the evidence in their favour, and possibly the transition probability between successive candidates. TRACE performs this same function in a connectionist system rather than by traditional programming methods. We can think of the initial bottom-up input as specifying the word lattice, and the final sequence of highly activated words as specifying a path through the lattice.

What we would like to do would be to find a way of combining the best properties of the recurrent network with the best properties of TRACE. That is, we would like to use only a single lexical network, but at the same time ensure that each segment of the input is only ever attributed to a single word, even when following context causes the initial interpretation to be modified.

One way to achieve this would be to use a recurrent network to generate a small set of lexical hypotheses – a short-list. These lexical hypotheses could then form the basis of a small interactive activation network which would perform the lexical competition. We can think of the recurrent net as generating a set of lexical candidates (the word-lattice) based purely on bottom-up information. No top-down feedback from later processes influences either phoneme recognition or generation of the candidate set itself. This small set of candidates then has somehow to compete with each other in order to determine the final parsing of the input. If we could construct a network like the lexical level of TRACE which only contained the candidates generated by the recurrent network, then we could avoid the problem of duplicating the lexical network. We would need two networks, but only one would be a full-blown lexical network generating lexical

candidates. The other would be a relatively small network to handle the right-context problem.

Given that the second network in such a scheme would have to contain different information at different times, depending on the candidates generated by the recurrent network, it would have to be programmable. The network would have to be wired differently for different inputs. We have to have a network that is rather different in character from most connectionist networks. Most networks have a fixed wiring scheme. Although the pattern of weights in connectionist networks often vary as a function of learning, all that changes in the short term is the pattern of activation evoked by different inputs. That is to say, such networks compute the same function irrespective of the input. In the present network the effective pattern of connectivity in the network also has to change on a short time scale. The connections in the network must be programmable so that part of the network can compute different functions when presented with different inputs.

Interactive activation networks are sufficiently powerful that we can use them to construct complex layered processing models like TRACE. In general, such networks can have any possible pattern of facilitatory and inhibitory connections between nodes. However, the constraint satisfaction system we require to carry out the lexical competition process has a very restricted architecture. Each node (lexical candidate) is connected to other nodes representing incompatible lexical hypotheses by means of bidirectional inhibitory links. The weights between competing units are symmetrical. This means that we could also perform the lexical competition process using other connectionist constraint satisfaction procedures such as a Hopfield net (Hopfield, 1982) or a Boltzmann machine (Hinton, Sejnowski, & Akley, 1984). Alternatively, of course, we could compute the best path through the set of lexical candidates using an algorithm such as dynamic programming or a related technique. These alternative ways of implementing the constraint satisfaction process will all produce similar results. However, the interactive activation algorithm used by TRACE is both simple and familiar and will be used in all of the simulations reported here. It is important to note that in the interactive activation network used here the only interaction is between word nodes at the lexical level. There is none of the between-levels interaction which is such a characteristic feature of both TRACE and the interactive activation model of visual word recognition. In the present model the direction of information flow between levels is always strictly bottom-up.

McClelland has suggested that some of the problems involved in duplicating lexical networks in TRACE can be overcome by using programmable networks (CIDs) of the form used in the programmable blackboard model of reading (McClelland, 1985, 1986a, 1986b). According to this suggestion, the hardwired lexical networks would be replaced by programmable networks whose connections were determined by a single central lexical network. All of the crucial lexical information is then represented in the single lexical network and the programmable networks are programmed as and when required. All of the lexical

information is stored in a single network so any learning that takes place can be restricted to this one network. Also, McClelland (1986a) argues that the programmable networks can have fewer connections than hardwired lexical networks, so the system is more economical.

However, McClelland's proposals solve only part of the problem. The Connection Information Distribution (CID) scheme described by McClelland is only a solution to the time-invariance problem. It tells us how to program a set of small lexical networks, so that we can recognise words starting at different times, but it does not tell us how to wire the inhibitory connections between overlapping words in different networks. It is these inhibitory connections which give TRACE its ability to deal with right-context. Programming the inhibitory connections between words in different lexical networks represents a rather harder problem than programming the networks themselves. Whether two words in different networks should have inhibitory connections depends on the conjunction of their positions and their lengths. The central module of a CID is not sensitive to position or length of a single word, let alone the conjunction of two words. Clearly, we don't want the CID to encode a complete matrix specifying, for all possible word pairs and all possible onset positions, how much inhibition there should be between them. This would be equivalent to listing all of the inhibitory connections between words in TRACE. To reduce the number of inhibitory lexical connections we need to abstract over length and onset position rather than encode inhibitory information in a permanent store. We need a mechanism to determine whether words overlap and to arrange for them to inhibit each other accordingly. CIDs do not provide such a mechanism.

The suggestion being made here is that a single programmable network is used simply to solve the right-context problem. The time invariance problem will be solved by a version of the recurrent network. That is, instead of programming word recognition networks, just program a network to perform the lexical competition that is performed by the top lexical layer of TRACE. Such an approach has the advantage that the network responsible for lexical competition need only contain as many nodes as there are candidates in the short-list. If we can get by with considering only a few candidates starting at each segment position, then it might be possible to keep the lexical network very small indeed.

4. The Shortlist model

The model being presented here assumes that a system similar to the recurrent network generates a set of candidate words which are roughly consistent with the bottom-up input. Each candidate word is then programmed into a lexical competition network working on the same principles as the lexical level of TRACE. However, for the sake of computational expediency and speed, two important simplifications are made. First, the process of using candidates to

program the lexical network is not performed by any clever piece of connectionist magic. Instead the model simply wires up the lexical network using conventional programming techniques. Second, the recurrent network has one undesirable property: with currently available computing resources a recurrent network would be prohibitively time consuming to train with a realistically large vocabulary. So the output of the recurrent network is simulated by an exhaustive search through a large machine-readable dictionary. Neither of these simplifications alter the final behaviour of the model, they just ensure that the final behaviour appears in seconds rather than years.

The model therefore consists of two stages. In the first stage an exhaustive lexical search derives a short-list of word candidates which match the input. In the second stage these candidate words are wired into the constraint satisfaction network so that overlapping words inhibit each other in proportion to the number of phonemes by which they overlap. As in TRACE, candidate words in the lexical competition network are organised according to their onset segment in the input. However, in the current model the only candidates considered are those for which there is some bottom-up evidence. Unless the number of candidates is limited the network could end up being as large as TRACE. In most of the simulations presented here the number of candidates which can be considered at each segment is therefore limited to 30. Later, simulations will be presented which specifically address the issue of the size of the candidate set.

If there are too many candidates at a given segment then the candidates with the lowest bottom-up activation are eliminated to make space for candidates with higher scores. That is, they are unwired from the network and the new candidates are wired in. The network spans a limited number of segments (currently the length of the largest word in the lexicon). If the network wiring were fixed, like TRACE, it would soon run out of space when receiving continuous input. However, as each new segment arrives the candidates starting at the oldest segment are unwired to make space for a new set of candidates. The wiring of the network therefore changes dynamically in response to changing input.

Bottom-up activation of each candidate word is determined by its degree of fit with the input. In the current version candidates are generated by an exhaustive search of a machine-readable dictionary. All of the simulations reported here use a 6000 word subset of the CELEX database compiled at the Max Planck Institute for Psycholinguistics. However, similar results have been obtained with two other dictionaries, one of which has 26 000 entries.

5. The lexical search procedure

TRACE has to have a complete lexical network associated with each phoneme in the input. The present model has to have a set of candidate words associated

with each input phoneme. This small set of candidates performs exactly the same job as the full lexical network in TRACE. The candidates represent words with onsets at that segment position where the match between the input and the candidate exceeds some prespecified criterion.

The lexical search procedure is designed to simulate the behaviour of a large recurrent network in generating candidates in a purely bottom-up fashion. The parallelism of the recurrent network has to be simulated by exhaustive serial searches of a dictionary. As each new phoneme is presented to the model a complete lexical search is performed spanning the input up to N phonemes back, where N is the size of the largest word in the dictionary. Obviously, if the longest word in the dictionary is, say, 15 phonemes long, then new bottom-up input can never change the scores of words in the candidate set starting on the 16th phoneme back. As each new phoneme arrives the complete search and match process has to be performed to revise the bottom-up match scores for words with onsets at any of the previous N phoneme positions. The process has both to update the scores for existing candidates and determine whether any new candidates should be added to the sets.

For each word in the lexicon the search procedure computes a score representing the degree of match between the word and input for each segment where the word might start from. In most simulations reported each word scores +1 for each phoneme that matches the input and −3 for each phoneme that mismatches. So, if the current input is /k/, /æ/, /t/, "cat" and "catalog" will both score 3 (1 + 1 + 1) whereas "cap" and 'captain' will only score −1 (1 + 1 − 3).[4] The relative weighting of match and mismatch information has an important influence on the model's behaviour and will be addressed in later simulations.

This scoring procedure has the merits of simplicity and efficiency. However, a more realistic procedure would almost certainly be able to make some allowance for the possibility that failures in the phonemic analysis might result in insertions or deletions of segments. With the present scoring method insertion or deletion of a phoneme in the middle of a word will always result in a negative score.

Note that the use of a mismatch score can be considered analogous to the effect of adding inhibitory phoneme-to-word connections to TRACE. The mismatch score helps to restrict the number of candidates the model needs to deal with, but,

[4] Note that the output of the search procedure should be equivalent to running a version of TRACE with phoneme–word inhibition and no top-down connections or lateral inhibition and then creaming off output words with greater than a given degree of activation.

This simple scoring procedure actually produces some candidates that are unlikely to be produced by the recurrent network. Words embedded at the ends of other words (deride) will score as highly as words embedded at the beginning (riding). In line with the psychological data the recurrent net shows less activation for words embedded at the end of other words. The bottom-up score could be modified to reflect this fact, but as the competitive network actually produces the same behaviour anyway the simplest possible bottom-up score was used.

Table 1. *The operation of the search process. As each new phoneme arrives the lexicon is searched for words beginning with that phoneme and previous searches are updated*

Input	Search on words beginning
1. k	k
2. æ	æ + kæ
3. t	t + æt + kæt

as will be discussed later, it also turns out to be necessary to account for recent empirical data (Marslen-Wilson, Gaskell & Older, 1991).

For a word to be included in the candidate set it needs to have a bottom-up score greater than some preset criterion. Currently words more than one phoneme in length enter the set if they have a score greater than 1. Single phoneme words enter the set if they have a score of 1. Once a word is included in the candidate set it stays there unless it is displaced by a higher scoring word, even if its bottom-up score drops below criterion.

Table 1 shows an example of the operation of the search procedure.

6. The lexical network

Each candidate that the lexical search generates is wired into the lexical network. The most important feature of the lexical network is that words which receive support from the same section of the input must compete with each other. With few exceptions each phoneme should only be part of a single word. This means that overlapping lexical candidates must be connected together by inhibitory links. The weights on the inhibitory links are simply proportional to the number of phonemes by which the candidates overlap. The greater the overlap, the greater the inhibition. The pattern of inhibitory links between a subset of the candidate words generated from the input /kætəlɒg/ is shown in Fig. 2. For clarity the figure only shows the wiring of nodes which fully match the input. Candidates such as *battle*, which only partially match the input, are not shown.

The bottom-up activation of each candidate node is a product of the bottom-up score for that candidate and the bottom-up excitation parameter. The lexical network is an interactive activation network and functions in exactly the manner described by McClelland and Rumelhart (1981). The full set of model parameters is listed in the Appendix.

In summary then, the model operates as follows: As each new phoneme is presented to the model the lexical search procedure first updates the candidate

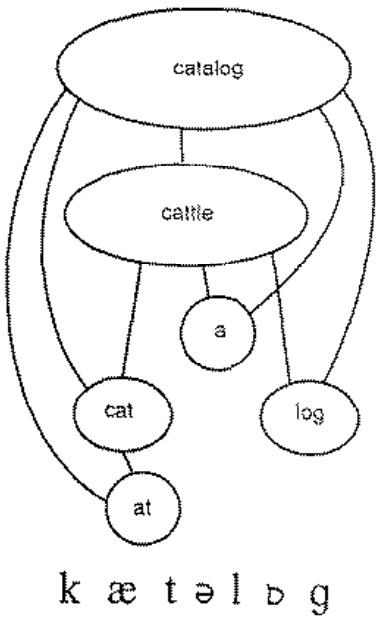


Figure 2. *The pattern of inhibitory connections between candidates produced by presentation of /kætəlɒg/. The figure shows only the subset of candidates completely matching the input. The full candidate set would also include words such as battle, catalyst, etc.*

sets and rewires the network as necessary.[5] The lexical search also updates the bottom-up activation score for each candidate. The lexical competition network then cycles through a fixed number of cycles (15 cycles in all of the present simulations) before the next phoneme is presented.

7. Comparison with other models

The Shortlist model shares a number of features with other models of spoken word recognition. In addition with its obvious similarities to TRACE, it also manages to capture many of the central insights of the Cohort model. In common with TRACE, Shortlist makes use of a competition mechanism to perform lexical segmentation. In common with the Cohort model there is a distinction between the initial bottom-up activation of potential word candidates (the cohort) and the subsequent winnowing down of the the cohort to identify a single word.

The model differs in a number of important respects from TRACE. First, the information flow in the model is bottom-up. That is, no stage in the model sends information back down to an earlier stage of processing. Most importantly, there is no top-down feedback from the lexical level to phonemic representations. The

[5] In a more complete implementation the bottom-up access procedure (search) would operate on the input continuously, updating the bottom-up scores progressively rather than producing new output only after each new phoneme.

bottom-up activation to the word nodes in the lexical network is determined solely by the degree of fit between the word and the input.

Words which mismatch the input have their bottom-up activation decreased. In the present model this is an essential move to help keep the candidate set size down to a manageable level. In TRACE there is no candidate set so there is no similar pressure to use mismatch information. In TRACE the mismatch score would be equivalent to having inhibitory connections between phonemes and words. All of the position specific phonemes nodes would have inhibitory connections to words which did not contain that phoneme in that position. McClelland and Elman discuss the possibility of using phoneme–word inhibition but decide against it, pointing out that the same effect is achieved by the word–word inhibition. If the input is /kæt/ then both *cat* and *cap* will be activated to some extent. However, there will be no need for /t/ to inhibit *cap* because lexical competition from *cat*, which will have higher activation, will inhibit *cap* anyway.

The use of mismatch information in TRACE is therefore lexically mediated. The input /kæt/ should inhibit *cap* because of the lexical competition between *cat* and *cap*. But /kæg/ should not inhibit either *cap or cat* because /kæg/ does not have a lexical node to generate inhibition. In a recent study, Marslen-Wilson et al. (1991) tested this prediction using a cross-modal priming paradigm. They found that any deviation of the input from the target word was sufficient to eliminate cross-modal priming regardless of whether the input was a word or a non-word. That is, while a word like /kæt/ might prime *dog*, both /kæp/ and /kæg/ would be equally ineffective in priming *dog*. With short non-word primes there is a possibility that there might never be sufficient lexical activation of the word to produce priming. That is /kæ/ might not activate *cat* much beyond resting level. However, they also found that word final mismatches failed to produce priming even with long words where the mismatch occurred after the uniqueness point. In these cases the non-word should have given rise to substantial lexical activation. For example, "apricod" failed to prime *fruit*. This lack of priming seems unlikely to be attributable to insufficient positive match information because splicing off the final phoneme and presenting subjects with "aprico" produced similar levels of priming to "apricot". According to TRACE, only the word competitor should eliminate the priming whereas the non-word should continue to prime, albeit to a lesser degree. The mismatching non-word and the truncated non-word should produce identical priming. In the current model the degree of priming will depend on the mismatch parameter. If the mismatch parameter is high then any deviation will greatly reduce the activation of a candidate. With a very low mismatch parameter a candidate may remain sufficiently strongly activated to still produce priming.

A comparison of Shortlist and the Cohort model is hindered by the fact that the Cohort model has no explicit computational formulation. In the most recent

expression of the Cohort model (Marslen-Wilson, 1987) the all-or-none nature of the original Cohort model has been tempered somewhat. Entry into the cohort now depends on some degree of goodness of fit between a lexical representation and the input rather than on an exact match. Recognition is no longer a matter of reducing the cohort to a single member, but now depends "on the process of mutual differentiation of levels of activation of different candidates" (Marslen-Wilson, 1987, p. 99). In fact, there is no longer even a distinct set of words that we can definitively state are members of the cohort. Many candidates may be momentarily activated but "it takes some amount of time and input for candidates to start to participate fully in the selection and integration process" (p. 99). It is this subset of active candidates which effectively constitute the word-initial cohort. This set of candidates would seem to correspond closely to the members of the candidate set in the Shortlist model. However, in the present model the functional distinction between those words which are considered to be members of the candidate set and other words is directly reflected in the architecture: candidates, or members of the cohort, are given a representation in the candidate sets which is distinct from the representation derived from the bottom-up access procedure. Note that this is not simply an implementational concern. The need to select a candidate set is a necessary consequence of the need to deal with the problem of lexical segmentation without duplicating the lexical network. The issue of lexical segmentation is not one which the Cohort model directly addresses. However, the functional architectures of Shortlist and the Cohort model are very similar. Both begin with a data-driven process of candidate selection which makes use of a goodness of fit measure, and both use mismatch as well as match information to home in on a single candidate. This candidate need not be the only candidate in the set, but should have a higher level of activation than its competitors.

8. Relationship to the race model of Cutler and Norris

One of the major characteristics which distinguishes Shortlist from TRACE is the fact that Shortlist is a data-driven system. In Shortlist there is no top-down feedback from the lexical level to the phoneme level. However, top-down feedback is an essential component of the account TRACE gives of a number of phenomena ranging from lexical influences in phoneme monitoring to sensitivity to phonotactic constraints. Without this top-down feedback how can Shortlist account for these important phenomena? The explanation given by Shortlist for these phenomena is precisely the same as the explanation given by the race model. In fact, we can think of Shortlist as being an implementation of exactly the kind of lexical access system envisaged by the race model.

The empirical evidence against TRACE comes largely from studies contrasting

the interactive predictions of TRACE with the predictions of bottom-up models such as the race model of Cutler and Norris. In the race model there are two ways of performing phoneme identification, one based on a purely bottom-up phonological analysis of the input, the other based on reading phonological information out of the lexical representation. The two routes race against each other and the final response is determined by the first route to produce an output. Shortlist shares two basic features with the race model. First, the phonological analysis is completely autonomous. There is no top-down feedback from the lexical level to the phonological analysis. Second, the model has explicit phonological representations in the lexicon. Phonological information has to be read out from these representations in order to align lexical candidates with the input. If these representations can also be used for performing phoneme identification the model has all of the basic components of the race model. It has a phonemic route from the phonological input, and a lexical route from the lexical representations.

Because it is a race model Shortlist is able to explain all of the phenomena which have been cited as problematic for TRACE. For example, Frauenfelder, Segui and Dijkstra's demonstration that lexical effects on phoneme monitoring can only be facilitatory and not inhibitory is explained exactly as in the race model. The race model assumes that although the lexical route can produce faster responses than the phonemic route, no matter how much the lexical route is slowed this will have no impact on the speed of the phonemic route. Similarly, the accounts of the phoneme monitoring data of Cutler et al. and the Ganong effect data of McQueen are also identical to the race model explanations already discussed.

9. Shortlist as part of a larger modular system

As described so far, Shortlist does not have the same broad coverage as TRACE. Shortlist is intended primarily as a model of the lexical processes in word recognition. However, TRACE is a model of both word recognition and phoneme recognition. The interactive nature of TRACE dictates that phenomena at both the word and phoneme level have to be considered together. In TRACE, phoneme recognition, even in non-words, can be strongly influenced by top-down feedback from the lexical level. In Shortlist, the processing at the phoneme level is totally unaffected by lexical-level processing. Consequently, Shortlist can be considered to be a single component in a modular system (in fact it is two components, generation of the candidate set followed by the lexical competition process). The other component needed for a complete word recognition system is a phoneme recogniser. The assumption being made here is that the phoneme recogniser would take the form of the recurrent net described earlier. This is

clearly a very practical proposition because similar networks are already in use in automatic speech recognition systems. However, the advantages of such networks are more than just technological. They also have interesting properties as psychological models. The interesting psychological properties of such networks stem, once again, from the fact that the time-delayed connections give the net a memory for its previous processing operations and therefore enable it to integrate information across time. So, for example, such networks readily learn to cope with the effects of coarticulation (Norris, 1992, 1993). The categorisation of each phoneme can be contingent on the identity of the previous phoneme. In TRACE such compensation for coarticulation is achieved by hard-wiring top-down connections from the phoneme to the featural level. In a recurrent network proper treatment of coarticulation is an automatic consequence of the fact that each phoneme is processed in the context of a memory for prior processing operations. The sensitivity to prior phonemic context exhibited by recurrent networks can extend across a number of intervening phonemes to make the network sensitive to statistical and phonotactic properties of the input.

For example, Norris (1993) simulated Elman and McClelland's (1988) results using a recurrent net identical to that described in Fig. 1. The network exhibits categorical perception, compensation for coarticulation (Mann & Repp, 1981; Repp & Mann, 1981) and the Ganong effect. Elman and McClelland's results are simulated by combining these three effects. The network was trained to identify the current phoneme in the input and to anticipate the next. After training on a set of three-phoneme "words", the network showed a bias to interpret ambiguous word-final phonemes so as to form a word rather than a non-word. The net had learned the statistical regularities in its input so that after encountering the first two phonemes in a word it would fill in the final phoneme even if no input was ever presented. The network therefore displayed the Ganong effect by developing a sensitivity to statistical sequences in the input rather than by allowing any top-down flow of information from a lexical level to influence phoneme processing. Because the network simply did not have any higher level nodes trained to identify words there was no possibility of a top-down lexical influence on processing. Shillcock, Lindsey, Levey, and Chater (1992) have extended this work and shown that the results of these simulations hold even with a vocabulary of 3490 words. So, in the Shortlist model there are actually two potential sources of the Ganong effect. Lexical effects may have their origin entirely within the lexical level as in the race model explanation of top-down effects, or they may have their effect entirely within the phoneme level due to the phoneme level developing sensitivity to statistical regularities in the input.

The performance of recurrent network phoneme recognisers, where sensitivity to phonotactic constraints and other statistical properties of the input develops entirely within the phoneme level, contrasts with the top-down explanation of such phenomena offered by TRACE. In TRACE, sensitivity to phonotactic

constraints arises because lexical feedback favours inputs which obey phonotactic constraints more than it favours inputs which violate such constraints. The explanation in terms of recurrent networks therefore represents a return to a more traditional style of modular linguistic explanation in which phonotactic knowledge is embedded in the phonological system itself, rather than being implicit in the structure of words in the lexicon. The irony here is that the top-down TRACE explanation would probably only have occurred to someone working within a connectionist framework. However, as we expand our connectionist horizons to include systems that learn, we find that they give a natural expression to the earlier and more modular theories.

The contrasting accounts of phonotactic effects given by TRACE and a bottom-up theory such as Shortlist have some interesting implications in interpreting data showing that lexical effects in phoneme monitoring may be modulated by attentional and task factors (Cutler et al., 1987; Eimas, Hornstein, & Peyton, 1990; Eimas & Nygaard, 1992). Where no lexical effects are observed, this implies that there is little or no top-down activation in TRACE. If there is no top-down activation there should be no phonotactic effects either, since they also depend on top-down activation. In the Eimas and Nygaard study, no lexical effects were observed when the target phonemes appeared in sentential context, but lexical effects were obtained when the words appeared in random word contexts in conjunction with a secondary task. If top-down effects are absent in sentential context then, according to TRACE, phonotactic effects should be absent also. This leads to the strange prediction that phonotactic effects should be absent in the conditions corresponding most closely to normal comprehension. According to the model being presented here, phonotactic effects are due entirely to operations within the phonemic level. Whether or not lexical effects are present is determined by whether subjects attend primarily to the lexical or phonemic levels. The phonemic level should continue to use phonotactic constraints regardless of where subjects attention is directed. Also, the strongest phonotactic effects should therefore be observed when responses are determined predominantly by the phonemic level and lexical effects are at their weakest.

We can now see that, although Shortlist is primarily concerned with lexical processes, it fits into a larger modular framework that gives a broad coverage of a wide range of data on human speech recognition. As an implementation of the race model of Cutler and Norris Shortlist inherits the ability to explain a range of effects which appear to demonstrate an influence of lexical information on phoneme identification. At the phonemic level, the work with recurrent networks shows that lower level phenomena like categorical perception, and sensitivity to phonotactic constraints can also be accounted for within this modular architecture and that phonemic processing remains completely unaffected by higher level lexical processes. In the simulations that follow we will see that Shortlist

complements this earlier work by showing how such a modular system can cope with the taxing demands of large vocabulary word recognition.

10. Input representation

The primary input to the model consists of a string of phonemes. The choice of input representation was determined largely by the simple practical consideration that this is the form of representation used in most machine-readable dictionaries. However, we also wished to be able to study the behaviour of the model with less than perfectly resolved input. This was achieved by allowing the model to accept a mid-class phonetic transcription (Dalby, Laver, & Hiller, 1986). The mid-class transcriptions are used to represent a degree of uncertainty, or ambiguity in the input. There are 12 mid-class categories, each of which corresponds to a small class of phonemes such as voiced-stops. Phonemes within a mid-class category tend to be highly confusable with each other (Miller & Nicely, 1955; Wang & Bilger, 1973). The full set of mid-class categories is shown in Table 2. Each mid-class category is assumed to match all phonemes in the class equally well, but to mismatch all other phonemes. The match score for mid-class phonemes is set to 0.7 of that for a full match.

In addition to a full set of phonemes there is also a silence, or word boundary symbol. The symbol for silence mismatches all phonemes. There is also a symbol for noise. This symbol is designed to represent noise of a sufficient intensity to

Table 2. *Mid-class categories from Dalby et al. (1986)*

voiced stops
voiceless stops
strong voiceless fricatives
weak voiceless fricatives
strong voiced fricatives
front vowels
back vowels
central vowels
diphthongs
nasals
liquids
glides

mask any speech input. The noise symbol therefore neither matches nor mismatches any phoneme.

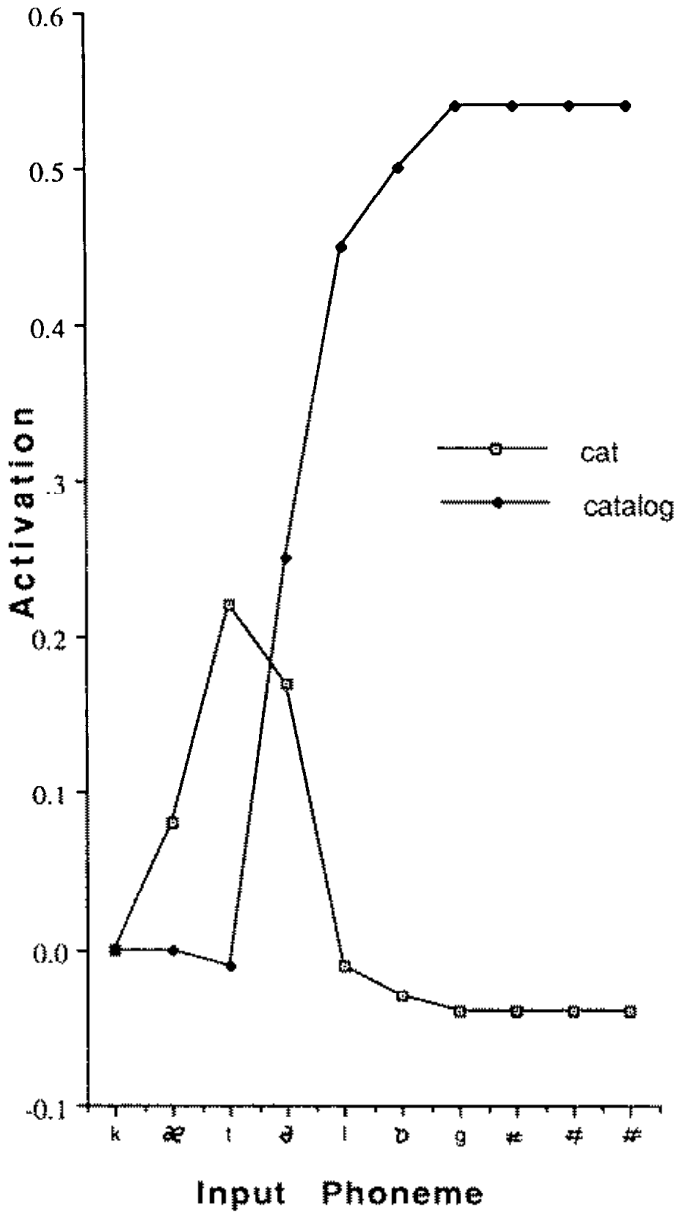
11. Simulations

The central motivation behind the present model is to provide a solution to the right-context problem faced by the simple recurrent network. So the first question we have to ask is whether the model can successfully handle cases like *catalog*. Figure 3 shows the model's behaviour when presented with /kætəlɒg/ as input.

In this figure, as with all other representations of the model's output in the paper, the lines indicate the activation level of particular candidate words following the presentation of each successive phoneme. The starting segment of the candidate words is never indicated because this can always be readily deduced from the input. For example, /kæt/ and /kætəlɒg/ must clearly both be members of the candidate set beginning at the phoneme /k/. The graph simply indicates how the activation of these two words in the /k/ candidate set changes over time.

The output of the model is exactly as one would have hoped. Initially *cat* is slightly more activated than *catalog*, but once the /ə/ arrives the situation reverses until finally *catalog* completely suppresses *cat*. The initial advantage for *cat* over *catalog* is due to the fact that *cat* is shorter. Inhibition between words is proportional to the number of phonemes by which the words overlap. Long words have more overlap with other long candidates and therefore receive more inhibition than short words. TRACE exhibits the same behaviour for exactly the same reasons.

The case of processing words containing other embedded words is only a single example of the importance of taking right-context into account in spoken word recognition. A number of studies have now demonstrated that information presented after word offset can have an important role in word recognition (Connine, Blasko, & Hall, 1991; Cutler & Norris, 1988; Bard, Shillcock, & Altmann, 1988; Grosjean, 1985). For example, Bard, Shillcock, and Altmann used a gating task to show that 21% of words successfully recognised in spontaneous speech are not recognised until well after their offset. The study by Connine et al. attempted to estimate the time span over which right-context can continue to exert an influence on earlier processing. They had subjects identify words whose initial phoneme was phonetically ambiguous (TENT/DENT). Semantic context was found to influence the categorisation of the ambiguous word if it arrived within three syllables of its offset but not if it was delayed until six syllables after the word's offset. Taken together these studies make it clear that any satisfactory model of human speech recognition must have the flexibility to keep its options open for several phonemes so that early decisions can be modified by later context.



Figue 3. *Activation of the words "cat" and "catalog" when the model is presented with the input "catalog". Note that words shown as having zero activation are not in the candidate set.*

12. Mismatch evidence

In order to keep the candidate sets down to manageable proportions we need to assume that any mismatch between a candidate and the input will reduce the candidate's bottom-up score. Mismatch information can also act to speed recognition of the best fitting candidate. Candidates that do not fit the input will

rapidly have their activation reduced and will therefore present less competition for the winning candidate. It also turns out that mismatch information is important to account for the results of the study by Marslen-Wilson, Gaskell, and Older. The question that arises though is what the relative weighting of match and mismatch information should be. If the weighting of mismatch information is too small then it will not do its job of keeping down the size of the candidate set and speeding recognition. If it is too big then any small errors in the pronunciation of a word will reduce the bottom-up score so much that the word will not be recognised. With a very large weighting of mismatch information the model would behave rather like the first version of the Cohort model. Words would effectively drop out of the candidate set as soon as the input deviated in any way from the expected form of the word. Norris (1982) pointed out that this was a major problem for the original Cohort model. If the Cohort model heard *cigarette* pronounced "shigarette" it could not possibly be recognised because *cigarette* would never be incorporated into the cohort. The same problem would arise if the initial phoneme of "cigarette" were masked by noise. The ability of the present model to overcome these problems is assessed in the next simulation which investigates the relative importance of match and mismatch information.

Figure 4 shows the growth of activation of *cigarette* during the presentation of /sɪgərɛt/, /ʃɪgərɛt/ and /?ɪgərɛt/, where /?/ represents noise of sufficient intensity to mask any phoneme present. Remember that such noise neither matches nor mismatches any phoneme. In this simulation the match parameter is set to 1.0 and the mismatch parameter to 3.0. Both of the distorted inputs reach a relatively high level of activation indicating that the model is not overly sensitive to small deviations in the input. Note that /ʃɪgərɛt/ produces less activation than /?ɪgərɛt/ because the /ʃ/ is a mismatch to *cigarette* whereas the noise neither matches nor mismatches.

Table 3 shows the final level of activation after presenting /sɪgərɛt###/, /ʃɪgərɛt###/ and /?ɪgərɛt###/ for increasing values of the mismatch parameter and the activation level after /sɪgə/. It can be seen that although increasing the mismatch parameter has no influence on the final level of activation of *cigarette*, by reducing the impact of competitors, it does help the word to be recognised earlier. Increasing the value of the mismatch parameter beyond about 3.0 simply serves to make the model more sensitive to slight distortions in the input. Although there is a tension between the requirement to keep the mismatch parameter high to reduce competitors and keeping it low to avoid oversensitivity, fortunately there is a middle ground where we can get most of the benefits of a high mismatch parameter without making the model too sensitive to small distortions of the input. In this case, the model gives us the best of both worlds. On the basis of this simulation the mismatch parameter is set to 3.0 in all of the remaining simulations.

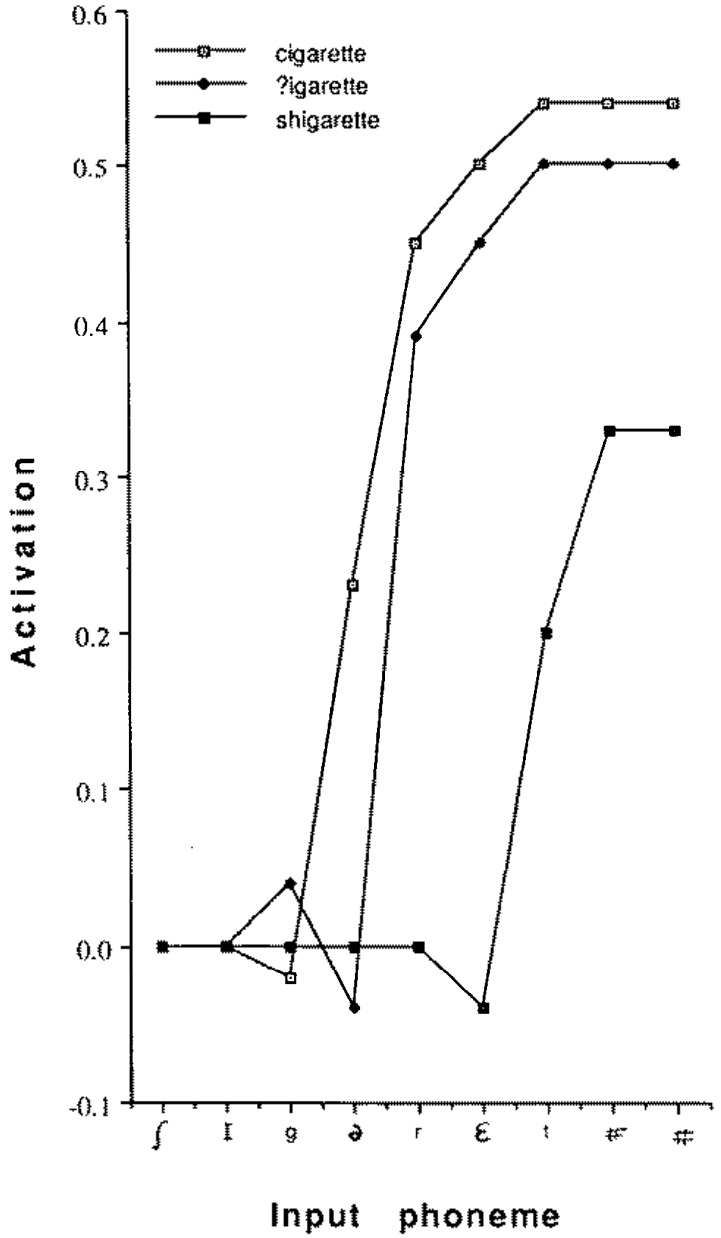


Figure 4. *Activation levels of "cigarette" following presentation of "cigarette", "shigarette" and "?igarette" where "?" represents noise of sufficient intensity to mask the initial phoneme completely.*

13. Ambiguity

The next simulations address the issue of how the model copes with ambiguous input. This is examined by replacing at least one of the phonemes in a word by a

Table 3. *Activation after three periods of silence beyond end of word*

Mismatch	1	2	3	4	5
/sɪgəret###/	0.54	0.54	0.54	0.54	0.54
/ʃɪgəret###/	0.45	0.40	0.33	0.25	−0.10
/ʔɪgəret###/	0.50	0.50	0.50	0.50	0.50
Activation after /sɪgə/					
Mismatch	1	2	3	4	5
/sɪgəret###/	0.21	0.23	0.23	0.24	0.32

mid-class transcription. The mid-class transcription can be considered to be a phoneme whose analysis has not been fully resolved. The input is still compatible with a small set of phonemes sharing a number of phonetic features. Comparison of activation levels for the original fully transcribed word and the same word with some phonemes replaced by mid-class transcriptions will give us some idea of how robust the model is when presented with an imperfect input. Remember that the 6000-word lexicon employed here is 30 times the size of the lexicon employed in TRACE simulations. Input containing some phonemes transcribed at the mid-class level might therefore be expected to generate a very large number of spurious lexical candidates which could severely impair the model's performance.

Two simulations are presented, one using words three phonemes long, the other using words six phonemes long. Each word is presented to the model preceded by one randomly selected context word and followed by two others. There were 50 three phoneme words and 50 six phoneme words. Each word was presented once with a full transcription, once with the first phoneme given a mid-class transcription, and once with the final phoneme given a mid-class transcription.

Figure 5 shows the mean activation levels for the three phoneme words plotted from the initial phoneme through to the fifth phoneme following the word. Figure 6 shows the mean activation levels for the six phoneme words plotted from the initial phoneme through to the sixth phoneme following the word. Both figures also show the average activation level of the strongest competitor to the presented word.

Not surprisingly, the longer words are clearly more resistant to degradation of a single phoneme than are the short words. However, even in the case of the short words, the final activation level of the mid-class words is over ten times that of the nearest competitor. The model is performing so efficiently here that its behaviour is effectively a reflection of the properties of the selection of words in its lexicon. Clearly this level of performance is only possible because the density of words in

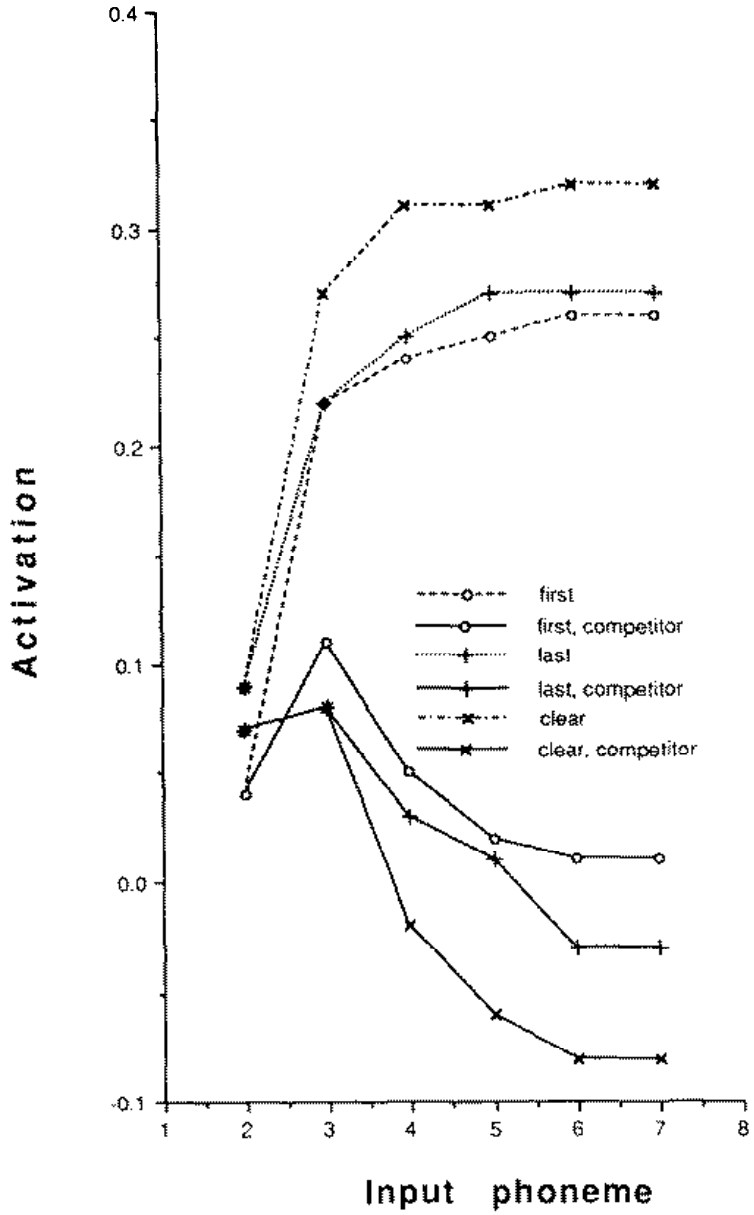


Figure 5. *Average activation levels of a three phoneme word and its nearest competitor when the word is clear or has either the first or last phoneme replaced by a mid-class transcription.*

the lexicon is not so great that neighboring words become indistinguishable when their first or last phonemes are given a mid-class transcription.

14. Continuous speech

The previous simulation involved recognising words embedded in context. The next simulation demonstrates the performance of the model on strings of words

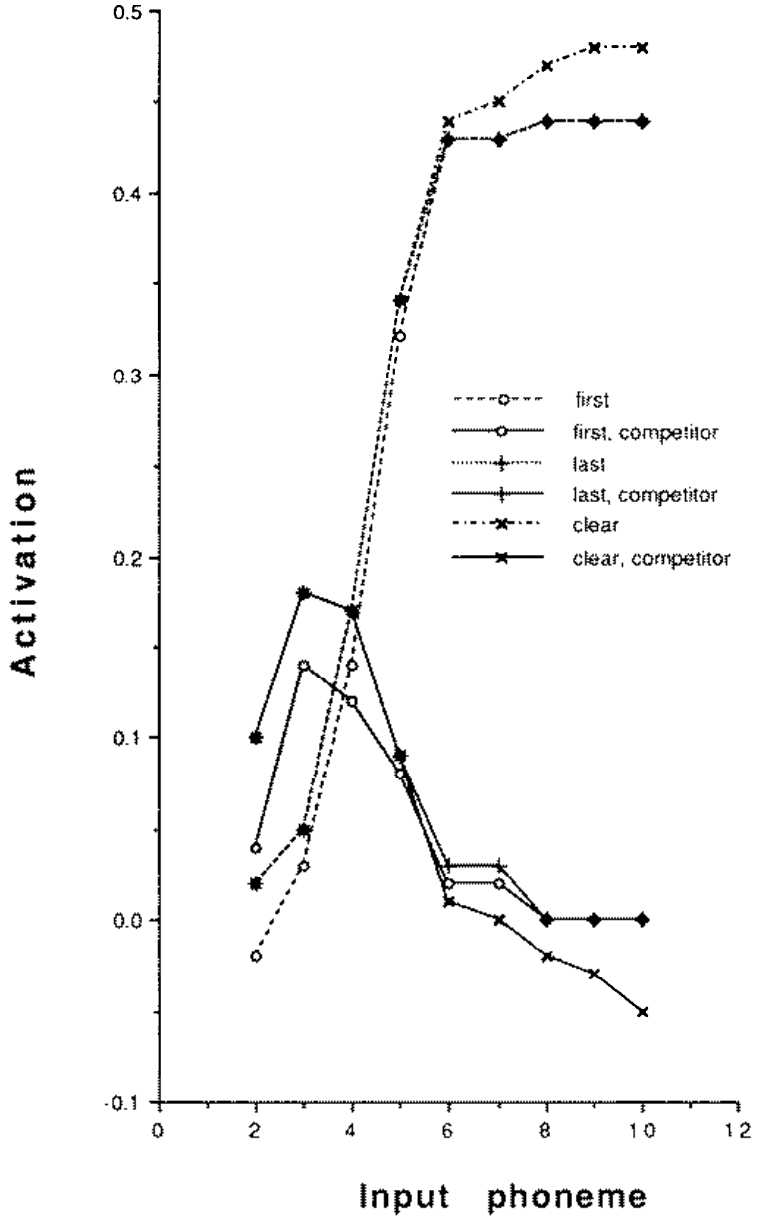


Figure 6. *Average activation levels of a six phoneme word and its nearest competitor when the word is clear or has either the first or last phoneme replaced by a mid-class transcription.*

specifically chosen because they contain other words embedded within them. Clearly such cases provide a strong test of the effectiveness of the competition mechanism with a large lexicon. Figure 7 shows the output of the model given the input "holiday weekend". As before, the graph does not indicate the starting segment for each candidate, but this should be clear from the input. By the end of the input the model has clearly resolved any temporary ambiguity in the analysis of the input and only the desired words remain at a high level of activation. Spuriously activated words have their activation suppressed below zero.

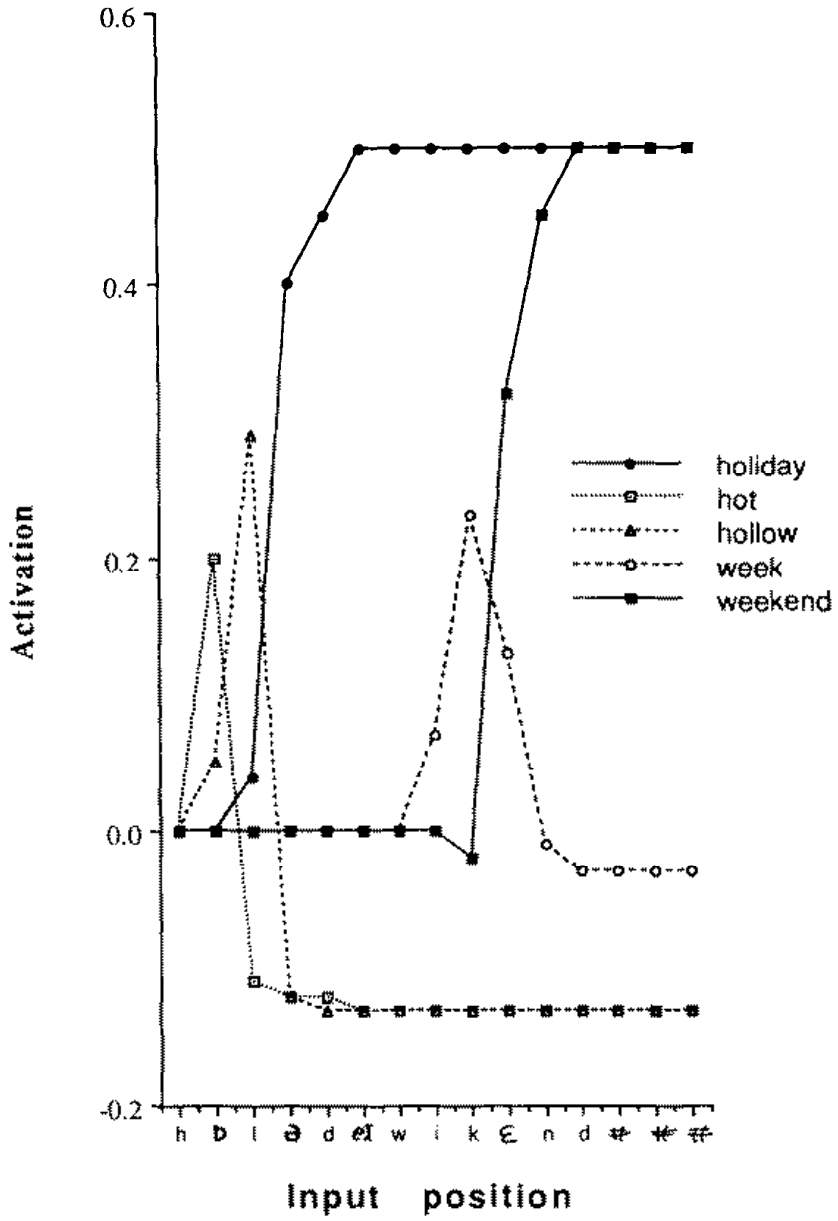


Figure 7. *Activation levels of words in the candidate set during the presentation of the input "holiday weekend". Note that words shown as having zero activation are not in the candidate set.*

Of perhaps greater interest, though, are cases where the interpretation of part of the input depends on information arriving several phonemes downstream. In the case of clearly transcribed input such examples appear to be rare. However, as ambiguity increases so the role of right-context will increase. The study by Bard, Shillcock, and Altmann demonstrated how, in a gating task using conversational speech, words were often only recognised correctly well after their offset. Figure 8 shows the model's response to the input /ʃɪpɪŋkwaɪəri/ (ship inquiry). In

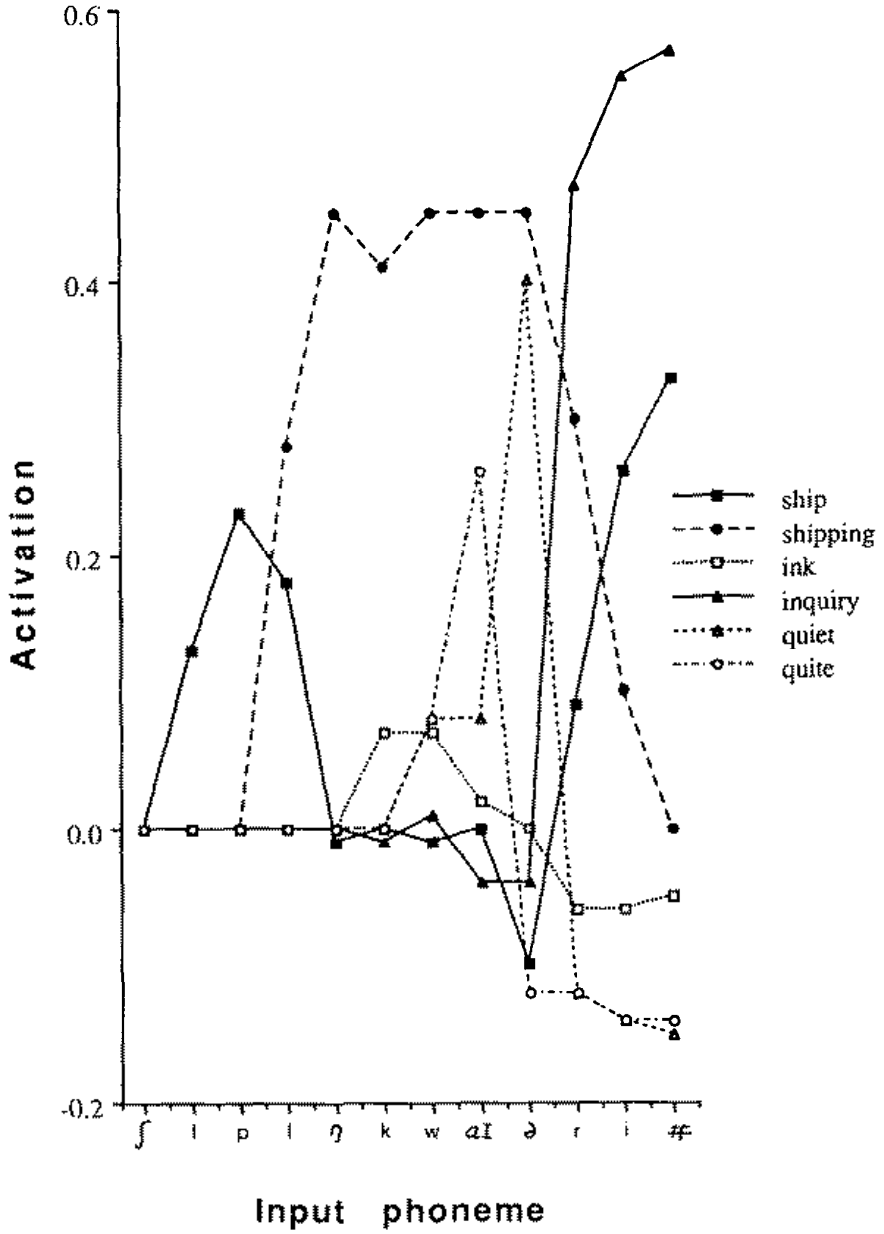


Figure 8. *Activation levels of words in the candidate set during the presentation of the input "shipping inquiry" with the limit on candidate set size at 30. Note that words shown as having zero activation are not in the candidate set. The example assumes British English pronunciation.*

this example correct parsing of the input can only be achieved well after the offset of *ship* when *inquiry* has been recognised. Before that *ship* is inhibited by *shipping* in much the same way that *catalog* inhibits *cat* in Fig. 3. Only when *shipping* itself is inhibited can *ship* win through.

In conjunction with the previous two simulations this demonstrates how the model remains remarkably robust even when the input is underspecified or potentially ambiguous for substantial portions of the input.

15. Candidate set size

In all of the simulations reported so far the candidate set for each segment has been limited to 30 words. The next simulation investigates the consequence of reducing the candidate set to its minimum size. The minimum size of the candidate set is two words. We need two words rather than one to deal with cases like "ship inquiry" where part of the input has a misleading analysis. The word *shipping* will always have a higher bottom-up score than *ship* because it contains more phonemes. Therefore, once shipping has made its way into the candidate set it can never be displaced by *ship*, even if it is strongly inhibited by *inquiry*. Entry into the candidate set is determined solely on the basis of bottom-up information. Therefore, in order to arrive at the correct interpretation of "ship inquiry" the set needs to be able to hold both *ship* and *shipping* as candidates simultaneously.

Figure 9 shows the results of processing 'ship inquiry' using a limit of only 2 candidates per segment. As can be seen, the activation levels are both qualitatively and quantitatively very similar in the two cases. The model works just as well with 2 candidates as with thirty.

With very degraded input the model will very likely need to consider more candidates. However, the important issue is not really whether the model will perform well with just two candidates per segment, but whether it can perform well with a very small number of candidates. To the latter question we can undoubtedly respond "yes". Note that in the present version of the model the available candidate nodes are permanently linked to a contiguous set of input positions. However, it will generally be the case that many segments will have no candidates at all because phonotactic constraints simply do not permit words to start with particular sequences of phonemes (in *tiptoe* there will be no candidates starting /pt/). Therefore only some fraction of the available candidate nodes will ever be used. If candidate nodes were allocated dynamically only to segments where they were needed then the total number of candidate nodes required could probably be halved. So, on average we may well require only one or two candidates per segment multiplied by the number of segments being actively considered.

In the example given earlier of TRACE operating with a 50 000-word lexicon in which all words were six phonemes in length, we need 550 000 word nodes and the lexical level alone of this network required over 10^{11} connections. Using, say, 10 candidates per segment the present network requires only 110 nodes and 4489

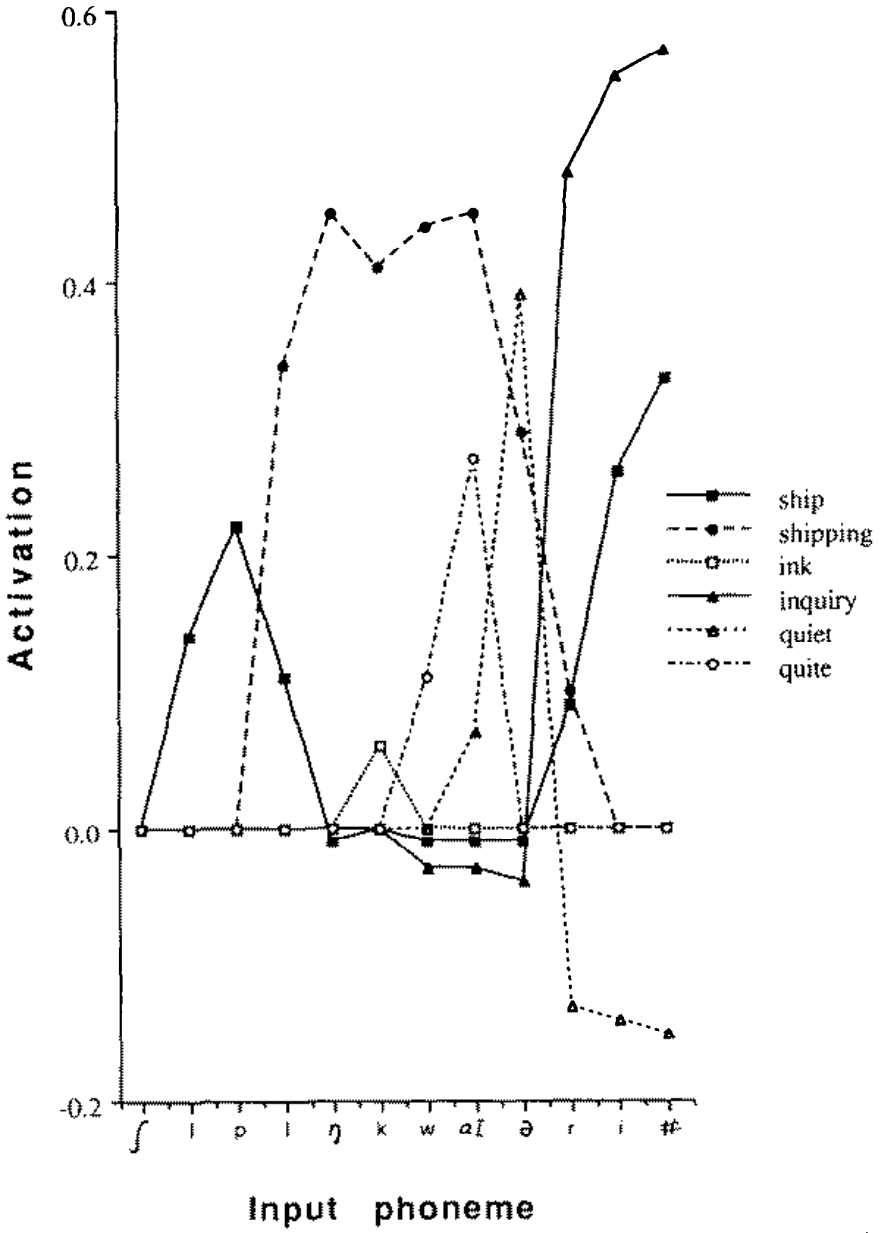


Figure 9. *Activation levels of words in the candidate set during the representation of the input "shipping inquiry" with the limit on candidate set size at 2. Note that words shown as having zero activation are not in the candidate set.*

connections. With 5 candidates per segment it would need only 1104 connections. So, with 5 candidates per segment the present model requires 10^8 times fewer inhibitory connections than the lexical level of TRACE.

16. Lexical competition

For their operation, both Shortlist and TRACE depend on competition between multiple lexical candidates. A large part of the present paper has been devoted to presenting a theoretical case that some form of lexical competition is an essential part of any system capable of recognising words in continuous speech. Nevertheless, the case for competition would be strengthened by empirical support for these ideas.

A number of studies have now used the cross-modal priming task to demonstrate that multiple lexical candidates are indeed activated during the early stages of auditory word recognition (Marslen-Wilson, 1987, 1990; Shillcock, 1990; Swinney, 1981; Zwitserlood, 1989). However, the fact that candidates are activated is no guarantee that they actually enter into competition with each other. More convincing evidence for both activation and competition comes from priming studies by Goldinger, Luce, and Pisoni (1989) and Goldinger, Luce, Pisoni, and Marcario (1992). These studies showed that recognition is inhibited when words are primed by similar sounding words. Inhibition of the primed word is assumed to be due to competition between the prime and the target. However, the most direct evidence of competition between lexical candidates comes from a study by McQueen, Norris, and Cutler (in press). They employed a word spotting task in which subjects had to identify words embedded in nonsense strings. Some of the nonsense strings were themselves the onsets of longer words. For example, subjects had to spot the word embedded in /dɔmɛs/, the onset of *domestic*, or in the nonword onset /nɔmɛs/. In three experiments McQueen et al. found that target words were harder to detect in word onsets like /dɔmɛs/ than in nonword onsets like /nɔmɛs/. The competitor word *domestic* appeared to be inhibiting recognition of the target. These competition effects persisted even when subjects knew in advance that target words would only ever appear at the ends of the nonsense strings. Under these circumstances subjects could, in principle, have ignored the onset of the competing word. So, not only do we have very solid evidence that lexical competition takes place, but it appears to be a mandatory process which subjects are unable to override even when the task would allow them to focus their attention away from the source of competition. McQueen et al. presented simulations to show that Shortlist gives an accurate account of the detailed pattern of competition effects observed in their experiments.

17. Processing prefixed words

With a small limit on the number of available candidates, prefixed words, or any words with very large initial cohorts, are going to pose problems. Even with

thirty candidates there is no guarantee that a word like *conductor* will make it into the short-list before the /d/. In the current implementation of the model the candidates are filled by a serial rather than a parallel search through the dictionary. This means that the members of a cohort listed first will enter the candidate set first and fill it up before later members are searched. However, remember that the ordered search is actually intended to simulate a parallel access process. So, a more plausible way of keeping the candidate set within limits might be to include only the most frequent words returned by the search.

Something rather interesting happens if a word makes a delayed entry into the candidate set because the cohort is initially much larger than the set size. Such a word must wait until its competitors are eliminated before gaining entry to the set. A word making a late entry will not have the same level of activation it would have had had it entered the candidate set at the earliest possible moment and been able to start building up activation. It then has to struggle with words which may now have a lower bottom-up score but which entered the set earlier and had more time to build up activation. A word making a late entry into the candidate set will therefore be recognised later than if it was included in the set from its onset.

One way to overcome this disadvantage that some words will suffer from is to make a single entry in the candidate set represent all words starting with the same string of phonemes. If the lexical search produces a large number of words which are identical up until the current phoneme then this set of words can be replaced by a single entry, or "cohort marker" in the candidate set. The cohort marker gets bottom-up input and fights with other candidates just like any single word candidate. However, when new input arrives which fits one member of the cohort better than others, this word inherits the activation level of the cohort marker. This best fitting candidate then behaves as though it had been in the candidate set right from the earliest point. Note that we might need several cohort markers in a single candidate set. As successive phonemes are presented they will deviate from some of the words represented by the first cohort marker. That is, not all words represented by the cohort marker will share the same initial cohort right up to the current input. At this point we will have to split the cohort marker. So, if the input is /kænd/, we might have one marker for /kænd..../, another for /kæns..../ and so forth. When a cohort marker gets split up, one of the new markers will necessarily have less bottom up evidence than another, so the lowest scoring marker can be dropped from the candidate set if there is insufficient space. Note that unless there is a misanalysis of the input, the correct word will always be represented by the highest scoring cohort marker and will therefore inherit the maximum amount of activation. The cohort-marker scheme has been implemented as an option in the present model and works well in overcoming the disadvantage of words in large word-initial cohorts. However, the cohort-marker option was not used in any of the simulations reported here.

Thus, although at first glance it appears that a network with only a small number of candidates per segment will have trouble processing large word initial cohorts, this is only a problem if all potential candidates have to be considered explicitly. If all of the words in a cohort can be represented by a single cohort marker then the candidate set can still be kept very small and this will have no disadvantages in terms of how quickly words can be recognised.

18. Lexical representations

The current model uses input that takes the form of a phonological representation of the input stream. Clearly the model could be modified to use featural or syllabic representations, or even to work from whole-word spectral templates. However, whatever form the input to the model takes, there must be an explicit form-based lexical representation of words expressed in the same vocabulary. The form-based representation is essential for the working of the model because the competition mechanism depends crucially on being able to align lexical candidates with the input. Each candidate has to know which section of the input it needs to stake a claim to. TRACE also has form-based representations to support the competition mechanism, although in TRACE these representations are implicit in the connections between the phoneme and lexical layers rather than being explicitly stored as part of a phonological representation in the lexicon.

This contrasts with the original recurrent network model which was simply a classification system. The recurrent network could produce a best guess as to what word was in the input, but it had no idea where the word began or ended. In this respect the recurrent network is rather like the logogen model (Morton, 1969). The network produces a response whenever it encounters a word but provides no information about the extent of the word in the input. A classification-only system might be perfectly adequate for phoneme recognition because phonemes never contain other phonemes as their constituents. But because words can contain other words as constituents, any effective word recognition system must be able to bind candidates to specific parts of the input stream.

In the majority of connectionist learning systems, such as those using back propagation, networks are simply trained to partition the input space. This means that such networks learn only as much as they need to in order to differentiate between the words they have been trained on. If a word can be recognised on the basis of its first few phonemes then the network may simply ignore the identity of subsequent phonemes. As a consequence, learning a new word can sometimes involve relearning a large part of the existing lexicon. This is because the network never really learns about the form of the words in the lexicon, it just learns how to tell them apart, and that ability may need to be based on completely new information if new words are added to the lexicon. For example, if a word has few

lexical neighbours it might be possible to identify it on the basis of a very superficial analysis of the input. If the lexicon grows, and the word acquires several close neighbours, a completely new analysis procedure will now be required to differentiate the word from its new neighbours. However, if the word recognition system begins by learning a form-based representation (say a phonemic representation) which captures all of the phonemic distinctions in the language then the basic representations and analysis procedures will never need to change because of changes or increases in vocabulary. A further problem faced by networks that perform classification without reference to form-based representations is that they will be unable to detect errors in pronunciation. Although such networks can recognise words that are slightly mispronounced, unlike human listeners, they have no way of knowing how the mispronunciation deviates from the target word. They simply do not have a representation of the expected word form against which they can compare the input. Mispronunciations simply reduce the overall activation level for the word. These networks have no way of determining what causes the activation level to be lower than normal.

The fact that TRACE does have implicit representations of word form in the connections between words and phonemes helps it to solve the right context problem, but TRACE will also suffer from problems in detecting mispronunciations unless it is somehow possible to interrogate the pattern of connections to determine which phonemes should be active for a particular word. The pattern of phoneme activations will indicate which phonemes are present, but not which phonemes should be present. So TRACE must also incorporate an explicit representation of word form. At the very least the information which is implicit in the connections must be made explicit by providing a mechanism which can interrogate the top-down connections and compare that with the bottom-up input. In the current model a representation of word form is essential to align candidates up with the input. A recurrent network will generate candidate words, but a form-based representation must be consulted to discover where the words begin and end in the input. In order to work at all, Shortlist must have access to the kind of form-based representations required for mispronunciation detection. In TRACE, mispronunciation detection depends on making phonemic representations explicit. But, once these representations are made available for the purposes of mispronunciation detection, they could also be made available for other tasks like phoneme identification itself. If TRACE could identify phonemes on the basis of lexical representations then it would have incorporated the race model.

19. Context and relation to checking model

The present model operates by identifying a candidate set of words before sufficient bottom-up information is available to identify the input uniquely. This is

exactly the form of the perceptual system required by the checking model (Norris, 1986) to account for context effects in word recognition. In the checking model all context effects take place between the point at which a perceptually derived candidate set of words is produced and the point at which the combination of perceptual and contextual information leads a single lexical candidate to exceed the recognition threshold. Candidates generated by the perceptual analysis are checked to evaluate their plausibility in the current context. Recognition thresholds are increased for implausible words and decreased for more plausible words. In the current model we can think of the checking process as increasing the activation of plausible candidates and decreasing it for implausible candidates. In normal discourse, where only a small proportion of the words are highly predictable from the context, we would expect most of the valuable work to be performed by the inhibitory effects of reducing the activation levels of less plausible candidates. When implausible words have their activation levels reduced the more plausible candidates will suffer less from competitive inhibition and will therefore be recognised more rapidly. Contextual inhibition should therefore be seen as having a healthy, facilitatory effect on recognition of any words which are not implausible in their context.

20. Parameter sensitivity

Whenever a model has a large number of parameters we need to know how sensitive the behaviour of the model is to small changes in those parameters. We have already investigated the mismatch parameter and seen that it can be varied over a wide range without greatly altering the behaviour of the model. The same seems to be true of all parameters other than inhibition. Small changes in the value of the word-to-word inhibition can produce quite large changes in the model's behaviour.

One of the main effects of inhibition is to alter the bias against long words. Long words are at a disadvantage relative to short words because they will overlap with more competitors. Each of those competitors is a source of inhibition. So, in the *catalog* example in Fig. 3 *catalog* has a lower activation than *cat* after the /t/ despite the fact that both words have the same amount of bottom-up activation. But, with too much inhibition long words can actually become difficult to recognise. Too much inhibition can also act to give early decisions excessive momentum. Once a candidate becomes highly activated it can suppress all competition. Later context is then totally unable to build up the activation of competing candidates and alter earlier decisions.

Of course, if the level of inhibition is set too low, spurious competitors do not have their activation suppressed and the network is unable to do its job of producing an unambiguous parsing of the input. For example, with inhibition set

at 0.08 *ship* and *shipping* are both strongly activated in the *ship inquiry* example. With inhibition at 0.15 *shipping* is never activated above *ship*.

So, while we want to employ as much inhibition as possible to make the network produce clean, unambiguous output, this can have the undesirable side effect of making the network insensitive to right-context, and it was the need to give an account of right-context effects that provided the initial motivation for the model.

Perhaps we should just be grateful that there is a small range of settings of the inhibition parameter that does lead to satisfactory performance with a wide range of different inputs. However, it is possible to make a small modification to the model so that it continues to perform sensibly even with very large settings of the inhibition parameter. The central problem with using large amounts of inhibition is that candidates that develop a high level of activation early on suppress all competitors, even competitors that should ultimately win out. A simple way to overcome this is to reset the network at regular intervals. In effect this deprives the network of its memory and allows it to settle into a new and optimal interpretation of the input. The resetting operation could be performed after a fixed number of cycles of the network, or could possibly be synchronised with the arrival of each new phoneme. In either case, following the reset all candidates start again on an even footing. Under this regime long words will still suffer an initial disadvantage relative to short words but, as soon as a long word gets more bottom-up support than a short word, it will win out because the short word will no longer be starting from the higher level of activation carried over from earlier processing. Such a change generally makes very little difference to the behaviour of the model until the inhibition is set high. With activation reset at intervals, high levels of inhibition no longer prevent the recognition of long words or diminish the influence of right-context.

Two recent studies provide strong empirical support for the idea of resetting activation and also show how Shortlist can be extended to incorporate the Metrical Segmentation Strategy of Cutler and Norris (1988). Norris, McQueen, and Cutler (submitted) and Vroomen and de Gelder (submitted) investigated the relationship between the Metrical Segmentation Strategy and lexical competition. Cutler and Norris had used a word spotting task to show that identification of CVCC words like *mint* is harder when they are embedded in a strong–strong CVCCVC nonsense word like /mɪntɛɪf/ than in a strong–weak nonsense word like /mɪntəf/. According to the Metrical Segmentation Strategy this is because *mint* in /mɪntɛɪf/ is segmented at the start of the strong syllable. Identification of *mint* therefore involves combining information across a strong syllable onset. In the strong–weak string /mɪntəf/ there is no such segmentation and identification of the target is easier.

Norris, McQueen and Cutler showed that the effect of metrical segmentation (the difference between strong–strong and strong–weak strings) is modulated by

lexical competition and only emerges when there is a large number of competing lexical candidates beginning with the /t/ of the second strong syllable. Vroomen and de Gelder also confirmed that, for strong–strong strings, the greater the number of competitors beginning at the /t/the weaker the activation of the target word.

Both of these studies model their data using a version of Shortlist modified to incorporate the Metrical Segmentation Strategy of Cutler and Norris (1988). In this version of Shortlist the scoring procedure for the lexical match is modified to reflect the relationship between the lexical representation of the candidate and the metrical structure of the input. Candidates starting at a strong syllable onset are given a boost if they themselves have a strong onset. If there is a strong syllable onset in the input and the candidate is not lexically marked as having a strong onset at that point then the bottom-up score is reduced. So, *mint* has its score reduced in /mɪntɛɪf/ because /t/ is the onset of a strong syllable whereas in the lexical representation of *mint* the /t/ will not be marked as being a strong onset.

Without the reset the number of competitors has a negligible effect on recognition of the target word. At the final phoneme the target word generally has such a high level of activation that potential competitors are strongly inhibited and fail to have any impact of the activation of *mint* itself. However, when using the reset these word final competitors do have an effect on the activation of the target word. Even after the end of the target word the reset ensures that the target and its competitors all start from zero activation. The competitors can therefore influence the target before becoming inhibited themselves. This is particularly so when the target word has its bottom-up activation reduced by the Metrical Segmentation Strategy. Shortlist can therefore successfully simulate this interaction between segmentation and lexical competition, but this does depend crucially on resetting the activation after each phoneme. Norris, McQueen and Cutler show that the modified version of Shortlist retains its basic character and gives an improved simulation of the data from McQueen, Norris, and Cutler (in press).

21. Conclusion

The lack of architectural elegance in TRACE is largely due to the fact that, in order to achieve time invariance, the basic lexical network has to be duplicated many times. Each lexical network then has to be interconnected with inhibitory links. In contrast, the recurrent network can perform time-invariant recognition using a single lexical network with a simple and elegant architecture. A recurrent network of this form works perfectly well when recognising isolated words. Even if the network begins by making the wrong decision, the decision it makes at the end of the word is usually the correct one. However, in continuous speech, any

decisions the network makes may have to be revised in the light of subsequent context. There is no independent way of determining when a word has ended so we cannot simply wait and read the output of the network only at word endings. But, with only a single set of lexical output nodes, the recurrent network has no continuing representation of earlier decisions. Once a decision is made it cannot be altered. If the only failing of the recurrent network were that it failed to maintain a record of its decisions, this could be remedied by keeping copies of the activation of the output nodes. However, the system needs to be able to compare the merits of lexical hypotheses generated at different times. This comparison depends on knowing which phonemes in the input generate support for each of the lexical candidates. This in turn depends on having a form based representation of words.

If we attempt both to generate lexical candidates and to perform lexical competition in the same network it is impossible to avoid duplicating the entire lexical network in the way that TRACE does. Each time-slice of the competition system has to be capable of recognising every word in the lexicon. However, by separating the process of generating lexical candidates from the competition process, we can dramatically reduce the scale of the competition problem. The lexical competition network need only consider a small short-list of candidates generated by a bottom-up lexical access system. This leads to an enormous saving in the number of inhibitory connections required between lexical candidates.

The main aim of the present enterprise was to produce a model that would combine the best properties of both TRACE and the recurrent network model within the framework of a modular, bottom-up system. Because of the desire to build a model that could operate with a large lexicon we have had to sacrifice the ability to learn in favour of a sizable vocabulary. Nevertheless, the model has successfully demonstrated that the basic architectural principles are sound. The model copes with a large vocabulary and the problems of revising decisions in the light of following context in a completely bottom-up system that only ever has to consider a small number of lexical candidates at each possible starting segment. With an unambiguous phonemic input the model was shown to work well with as few as two candidates per segment. While TRACE considers the entire lexicon as candidates, this model need only consider a small fraction of the lexicon as candidates at any one time. Indeed, because of phonotactic constraints, some segments may not have any candidates at all.

As the size of the lexicon is increased the task of finding a unique interpretation of a given input string obviously becomes harder. With a large lexicon there will be more embedded words and a greater potential for spurious lexical matches. However, this model continues to perform well when tested with a vocabulary of 6000 words and, as already mentioned, it continues to perform well even with a 26 000-word dictionary. Note that with a vocabulary of 26 000 words TRACE would effectively be considering all 26 000 words as candidates at all points!

Increasing the size of the vocabulary is one way of increasing the number of candidates that the model generates at each position. Another is by deliberately degrading the input by using mid-class transcriptions. Simulations using mid-class transcriptions once again showed the model to be very robust. Even replacing either the first or last phoneme word with a mid-class transcription resulted in less than a 9% decrease in the final activation level of the target word while still not allowing the average activation level of the nearest competitor to rise much above zero.

One of the central motivations behind the model was to overcome the problem of handling right-context faced by the recurrent network. In accord with this goal it was shown that the model could cope readily with input where local ambiguities temporarily lead the analysis up the garden path. The *ship inquiry* example demonstrates how the model can make use of context which does not become available until well after a word has ended. But, most importantly, all of this can be achieved using a very large lexicon, no top-down interaction, and as few as two lexical candidates per segment. However, the advantages of the Shortlist model are not just restricted to providing a more efficient and plausible architecture than TRACE. Shortlist also provides a better account of the data. Studies by Cutler et al. (1987), Eimas et al. (1990), Frauenfelder et al. (1990) and McQueen (1991a) all call into question the emphasis on top-down interaction that is such a central feature of TRACE. Instead, these studies all support a bottom-up autonomous model like Shortlist which embodies the basic architectural principles of the race model.

References

Bard, E.G., Shillcock, R.C., & Altmann, G.E. (1988). The recognition of words after their acoustic offsets in spontaneous speech: Evidence of subsequent context. *Perception and Psychophysics, 44*, 395–408.

Bellman, R. (1957). *Dynamic programming*. Princeton, NJ: Princeton University Press.

Burton, M.W., Baum, S.R., & Blumstein, S.E. (1989). Lexical effects on the phonetic categorization of speech: The role of acoustic structure. *Journal of Experimental Psychology: Human Perception and Performance, 15*, 567–575.

Burton, M.W., & Blumstein, S.E. (unpublished manuscript). Lexical effects on phonetic categorization revisited: The role of stimulus naturalness and stimulus quality.

Chien, L.F., Lee, L.S., & Chen, K.J. (1991). An augmented chart data structure with efficient word lattice parsing scheme in speech recognition applications. *Speech Communication, 10*, 129–144.

Connine, C., Blasko, D., & Hall, M. (1991). Effects of subsequent sentence context in auditory word recognition: Temporal and linguistic constraints. *Journal of Memory and Language, 30*, 234–250.

Cutler, A., Mehler, J., Norris, D., & Segui, J. (1987). Phoneme identification and the lexicon. *Cognitive Psychology, 19*, 141–177.

Cutler, A., & Norris, D. (1979). Monitoring sentence comprehension. In W.E. Cooper & E.C.T. Walker (Eds.), *Sentence processing: Psycholinguistic studies presented to Merrill Garrett*. Hillsdale, NJ: Erlbaum.

Cutler, A., & Norris, D. (1988). The role of strong syllables in segmentation for lexical access. *Journal of Experimental Psychology: Human Perception and Performance, 14*, 113–121.

Dalby, J., Laver, J., & Hiller, S.M. (1986). Mid-class phonetic analysis for a continuous speech recognition system. *Proceedings of the Institute of Acoustics, 8*, 347–354.

Eimas, P., Hornstein, S., & Payton, P. (1990). Attention and the role of dual codes in phoneme monitoring. *Journal of Memory and Language, 29*, 160–180.

Eimas, P., & Nygaard, L. (1992). Contextual coherence and attention in phoneme monitoring. *Journal of Memory and Language, 31*, 375–395.

Elman, J., & McClelland, J. (1986). Exploring lawful variability in the speech waveform. In S. Perkell & D.H. Klatt (Eds.), *Invariance and variability in speech processing* (pp. 360–385). Hillsdale, NJ: Erlbaum.

Elman, J., & McClelland, J. (1988). Cognitive penetration of the mechanisms of perception: Compensation for coarticulation of lexically restored phonemes. *Journal of Memory and Language, 27*, 143–165.

Frauenfelder, U., & Peeters, G. (1990). Lexical segmentation in TRACE: An exercise in simulation. In G.E. Altmann (Ed.), *Cognitive models of speech processing*. Cambridge, MA: MIT Press.

Frauenfelder, U., Segui, J., & Dijkstra, T. (1990). Lexical effects in phonemic processing: Facilitatory or inhibitory? *Journal of Experimental Psychology: Human Perception and Performance 16*, 77–91.

Ganong, W.F., III (1980). Phonetic categorization in auditory word perception. *Journal of Experimental Psychology: Human Perception and Performance, 6*, 110–125.

Goldinger, S.D., Luce, P.A., & Pisoni, D.B. (1989). Priming lexical neighbors of spoken words: Effects of competition and inhibition. *Journal of Memory and Language, 28*, 501–518.

Goldinger, S.D., Luce, P.A., Pisoni, D.B., & Marcario, J.K. (1992). Form-based priming in spoken word recognition: The roles of competition and bias. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*, 1211–1238.

Grosjean, F. (1985). The recognition of words after their acoustic offsets: Evidence and implications. *Perception and Psychophysics, 38*, 299–310.

Hinton, G.E., Sejnowski, T.J., & Akley, D.H. (1984). *Boltzman machines: Constraint satisfaction networks that learn*. Tech. Report No. CMU-CS-84-119, Pittsburgh, PA: Carnegie-Mellon Department of Computer Science.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA, 81*, 2554–2558.

Jordan, M. (1986). *Serial order: A parallel distributed processing approach*. ICS Report 8604. La Jolla: University of California, San Diego.

Mann, V.A., & Repp, B.H. (1981). Influence of preceding fricative on stop consonant perception. *Journal of the Acoustical Society of America, 69*, 548–558.

Marslen-Wilson, W.D. (1980). Speech understanding as a psychological process. In J.C. Simon (Ed.), *Spoken language understanding and generation*. Dordrecht: Reidel.

Marslen-Wilson, W.D. (1984). Function and process in spoken word recognition. In H. Bouma & D.G. Bouwhis (Eds.), *Attention and performance X. Control of language processes*. Hillsdale, NJ: Erlbaum.

Marslen-Wilson, W.D. (1987). Functional parallelism in spoken word-recognition. *Cognition, 25*, 71–102.

Marslen-Wilson, W.D. (1990). Activation, competition, and frequency in lexical access. In G.T.M. Altmann (Ed.), *Cognitive models of speech processing: Psycholinguistic and computational perspectives* (pp. 148–172). Cambridge, MA: MIT Press.

Marslen-Wilson, W.D., Brown, C., & Zwitserlood, P. (1989). *Spoken word recognition: Early activation of multiple semantic codes*. Unpublished manuscript, Max-Planck-Institute, Nijmegen.

Marslen-Wilson, W.D., Gaskell, G., & Older, L. (1991). *Match and mismatch in lexical access*. Paper presented at the spring meeting of the Experimental Psychology Society, Cambridge, April 1991.

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

Marslen-Wilson, W.D., & Zwitserlood, P. (1989). Accessing spoken words: The importance of word onsets. *Journal of Experimental Psychology: Human Perception and Performance*, *15*, 576–585.

Massaro, D. (1989). Testing between the TRACE and the fuzzy logical model of speech perception. *Cognitive Psychology*, *21*, 398–421.

McClelland, J. (1985). Putting knowledge in its place: A scheme for programming parallel processing structures on the fly. *Cognitive Science*, *9*, 113–146.

McClelland, J. (1986a). Resource requirements of standard and programmable nets. In D. Rumelhart & J. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition*, (Vol. 1). Cambridge, MA: MIT Press.

McClelland, J. (1986b). The programmable blackboard model of reading. In J. McClelland & D. Rumelhart (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2). Cambridge, MA: MIT Press.

McClelland, J., & Elman, J. (1986). The TRACE model of speech perception. *Cognitive Psychology*, *18*, 1–86.

McClelland, J., & Rumelhart, D. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of the basic findings. *Psychological Review*, *88*, 375–407.

McQueen, J.M. (1991a). The influence of the lexicon on phonetic categorisation: Stimulus quality and word-final ambiguity. *Journal of Experimental Psychology: Human Perception and Performance*, *17*, 433–443.

McQueen, J.M. (1991b). *Phonetic decisions and their relationship to the lexicon*. Unpublished PhD thesis, University of Cambridge.

McQueen, J., Norris, D., & Cutler, A. (in press). Competition in spoken word recognition: Spotting words in other words. *Journal of Experimental Psychology: Learning, Memory and Cognition*.

Miller, G.A., & Nicely, P.E. (1955). An analysis of perceptual confusions among some English consonants. *Journal of the Acoustical Society of America*, *27*, 338–352.

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

Norris, D. (1982). Autonomous processes in comprehension: A reply to Marslen–Wilson and Tyler. *Cognition*, *11*, 97–101.

Norris, D. (1986). Word recognition: Context effects without priming. *Cognition*, *22*, 93–136.

Norris, D. (1988). Paper presented at the Sperlonga Conference on Cognitive Models of Speech Processing, May 1988.

Norris, D. (1990). A dynamic net model of human speech recognition. In G.E. Altmann (Ed.), *Cognitive models of speech processing*. Cambridge, MA: MIT Press.

Norris, D. (1992). Connectionism: A new breed of bottom-up model. In R. Reilly & N. Sharkey, (Eds.), *Connectionist approaches to natural language processing*. Hove, UK: Erlbaum.

Norris, D. (1993) Bottom-up connectionist models of interaction. In R. Shillcock & G. Altmann (Eds.), *Cognitive models of speech processing: Sperlonga II*. Hove, UK: Earlbaum.

Norris, D., McQueen, J., & Cutler, A. (submitted). Competition and segmentation in spoken word recognition.

Pitt, M., & Samuel, A. (1993). An empirical and meta-analytic evaluation of the phoneme identification task. *Journal of Experimental Psychology: Human Perception and Performance*, *19*, 699–725.

Repp, B.H., & Mann, V.A. (1981). Perceptual assessment of fricative-stop coarticulation. *Journal of the Acoustical Society of America*, *69*, 1154–1163.

Robinson, A.J., & Fallside, F. (1988). A dynamic connectionist model for phoneme recognition. *Proceedings of the European Conference on Neural Networks*, Paris, 1988.

Rumelhart, D., & McClelland, J. (1982). An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, *89*, 60–94.

Shillcock, R. (1990). Lexical hypotheses in continuous speech. In G.T.M. Altmann (Ed.), *Cognitive models of speech processing: Psycholinguistic and computational perspectives* (pp. 24–49). Cambridge, MA: MIT Press.

Shillcock, R., Lindsey, Levey, J., & Chater, N. (1992). *A phonologically motivated input representation for the modelling of auditory word perception in continuous speech*. Paper presented at Cognitive Science 1992, Bloomington.

Swinney, D. (1981). Lexical processing during sentence comprehension: Effects of higher order constraints and implications for representation. In T. Myers, J. Laver, & J. Anderson (Eds.), *The cognitive representation of speech* (pp. 201–209). Amsterdam: North-Holland.

Thompson, H. (1990). Best-first enumeration of paths through a lattice: An active chart parsing solution. *Computer Speech and Language, 4*, 263–274.

Tomita, M. (1986). An efficient word lattice parsing algorithm for continuous speech recognition. *Proceedings of the 1986 IEEE International Conference on Acoustic Speech and Signal Processing*, pp. 1569–1572.

Vroomen, J., & de Gelder, B. (submitted). Metrical segmentation and lexical inhibition in spoken word recognition.

Waibel, A., Hanazawa, T., Hinton, G. Shikano, K., & Lang, K. (1988). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37.

Wang, M.D., & Bilger, R.C. (1973). Consonant confusions in noise: A study of perceptual features. *Journal of the Acoustical Society of America, 54*, 1248–1266.

Watrous, R.L., Shastri, L., & Waibel, A.H. (1987). Learned phonetic discrimination using connectionist networks. *In* J. Laver & M.A. Jack (Eds.), *Proceedings of the European Conference on Speech Technology*. Edinburgh: CEP Consultants Ltd.

Zwitserlood, P. (1989). The locus of the effects of sentential-semantic context in spoken-word processing. *Cognition, 32*, 25–64.

Appendix: model parameters

The model uses the interactive activation algorithm described in McClelland and Rumelhart (1981). Given that the model uses a network with only a single layer it has only the following 8 parameters. The parameters representing minimum and maximum activation simply scale the range of activation levels and are therefore not free model parameters in the sense that they have no effect on the pattern of behaviour exhibited by the model.

minimum word activation: −0.3
maximum word activation: 1.0
word-to-word inhibition: 0.12
bottom-up phoneme-to-word excitation: 0.05
decay: 0.3
score for a mid-class match: 0.7 of the match score
score for a mismatch: −3.0 times the match score
number of iterations through net per segment: 15