ELSEVIER

Cognitive Science 27 (2003) 807–842

COGNITIVE SCIENCE

http://www.elsevier.com/locate/cogsci

Two ways of learning associations

Luke Boucher, Zoltán Dienes*

Department of Psychology, Sussex University, Brighton, Sussex BN1 9QG, UK

Received 9 October 2002; received in revised form 3 March 2003; accepted 6 March 2003

Abstract

How people learn chunks or associations between adjacent items in sequences was modelled. Two previously successful models of how people learn artificial grammars were contrasted: the CCN, a network version of the competitive chunker of Servan-Schreiber and Anderson [J. Exp. Psychol.: Learn. Mem. Cogn. 16 (1990) 592], which produces local and compositionally-structured chunk representations acquired incrementally; and the simple recurrent network (SRN) of Elman [Cogn. Sci. 14 (1990) 179], which acquires distributed representations through error correction. The models' susceptibility to two types of interference was determined: prediction conflicts, in which a given letter can predict two other letters that appear next with an unequal frequency; and retroactive interference, in which the prediction made by a letter changes in the second half of training. The predictions of the models were determined by exploring parameter space and seeing how densely different regions of the space of possible experimental outcomes were populated by model outcomes. For both types of interference, human data fell squarely in regions characteristic of CCN performance but not characteristic of SRN performance.

Keywords: Learning associations; Chunks; SRN

1. Introduction

Human learning is often concerned with adapting to sequential regularities. Written language, spoken language and even behaviour in different social environments are skills we learn through experience that can be characterised by sets of complex rules concerning sequences of events. If connectionism can elucidate our understanding of learning in humans, then learning sequential structure is one area that it must be able to address. This paper

* Corresponding author. Tel.: +44-1273-678550; fax: +44-1273-678058.
E-mail address: dienes@biols.susx.ac.uk (Z. Dienes).

0364-0213/$ – see front matter © 2003 Cognitive Science Society, Inc. All rights reserved.
doi:10.1016/j.cogsci.2003.03.001

considers the behaviour of two connectionist models confronted with a simple experimental phenomenon: artificial grammar learning. Testing the models in this domain will be used to illustrate how connectionist models can have genuine explanatory power in understanding human learning. We will address the issue of whether it is useful for a learning model to acquire local representations of chunks of successive elements in a sequence, or whether chunking behaviour is more usefully regarded as an emergent property of other associative processes. Artificial grammar learning is simply taken to be an example domain where chunking is important, and hence a useful domain for investigating some general principles of human learning.

A seemingly ubiquitous and powerful type of learning in people is our capacity to chunk together elements into novel wholes (e.g., Anderson & Lebiere, 1998); it seems the chunk itself can then come to act as a familiar element in its own right. In reading, line segments come to be seen as letters, groups of letters as words, and groups of words as familiar phrases. In the same way, sets of actions can become well-rehearsed scripts, which could then become elements of larger scripts (Schank, 1982). Two approaches to modelling this pervasive style of learning are, firstly, to have a learning device that acquires localist representations of frequently co-occurring elements (e.g., Newell, 1990; Page, 2000; Perruchet & Vinter, 1998); secondly, to have distributed representations of the higher order structure emerge (e.g., Elman, 1990; Gaskell & Marslen-Wilson, 1999; Kinder & Shanks, 2001). An experimental paradigm that illustrates the learning of chunks is artificial grammar learning (Reber, 1989); indeed, at least a majority of the learning in this paradigm under the standard conditions used in the literature appears to consist of learning chunks of two or three letters (Dulany, Carlson, & Dewey, 1984; Johnstone & Shanks, 1999; Perruchet & Pacteau, 1990). Modelling in this domain has also exemplified both the localist and distributed approaches to modelling chunking, though up to now differential predictions of the models have not been developed and tested (Berry & Dienes, 1993). This paper will look at particular models inspired by the localist and distributed approaches, derive differential predictions and test them.

Initially a brief introduction to artificial grammar learning will be presented. Next the two connectionist models to be compared are described. Then we demonstrate there are basic differences in the way the two computational models learn and represent simple associations between pairs of items. The manner in which both people and models perform on two grammar-learning tasks devised to exploit these differences is then explored.

2. Artificial grammar learning

The artificial grammar learning paradigm was used by Reber (1967, 1989) as a means of eliciting complex and abstract learning without, on the one hand, a subject's explicit intention to learn nor, on the other hand, full awareness of what is learnt. He called this type of learning "implicit learning." Reber (1992) argued that implicit learning did not involve the conscious hypothesis—testing much investigated by psychologists at the time. In fact, Reber argued that implicit learning, in evolutionary terms, came first, and was thereby more robust and less variable than conscious hypothesis testing. How conscious people are of the knowledge is not an issue we will directly address in this paper, except in passing (see Berry, 1997; Cleeremans,

Destrebecqz, & Boyer, 1998; Dienes & Perner, 1999, 2002a, 2002b; French & Cleeremans, 2002, for discussion).

The standard artificial grammar learning experiment takes the following form (e.g., see Berry, 1997). There are two phases. In the first phase, subjects are warned of an impending memory test and asked to try and memorise a set of exemplars. These generally take the form of letter sequences, such as "MTVRX." Subjects are not informed of the existence of any underlying structure, or set of rules that might have been used to produce these stimuli, but the sequences are produced by a finite state grammar, determining, for example, the allowable letters to follow an "M" or whether an "R" can be repeated.

The second phase is a test. Subjects are informed of the existence of a complex set of rules that were used to determine the order of letters in the exemplars they have just seen. But they are not told what these rules are. Instead, subjects are asked to classify a new set of exemplars into those that obey these rules and those that don't. They are then presented a set of test exemplars, half of which are produced by the same set of rules and the remaining half of which are not.

In general, subject's classifications in this test phase reflect the fact that they have learned (or can generalise their responses to) something of the stimuli's underlying structure. In a standard classification test subjects produce non-random responses, classifying with above 50% accuracy. One of Reber's (1989) conclusions from these experiments was that since subjects could generalise their responses to new exemplars, they had actually learned the underlying abstract grammar that was used to produce the stimuli. Research since then has specified the contents of people's knowledge more precisely. One way of learning the structure specified by the grammars used would be to learn the statistical redundancy in the stimuli to progressively higher orders. Indeed, Perruchet and Pacteau (1990) showed that subjects simply exposed to grammatical bigrams in the training phase classified new test items very similarly to subjects exposed to complete grammatical strings: A large part of what people learn in the artificial grammar learning paradigm is bigrams. Perruchet and Pacteau (1990) and Dienes, Broadbent, and Berry (1991) extended these findings by showing that people could in addition become sensitive to trigrams. Since then, the importance of bigram and trigram knowledge has been shown by a number of investigators. Redington and Chater (1996) demonstrated that the learning of bigrams and trigrams was in principle sufficient to allow transfer between different domains, a phenomenon previously cited as problematic for the fragment account of artificial grammar learning. Moreover, Knowlton and Squire (1994, 1996) and Meulmans and Van der Linden (1997) showed that subjects' classification performance could be predicted from a measure of the frequency in the training set of the bigrams and trigrams in each test item. However, people's knowledge does not stop at learning bigrams and trigrams. There is systematic residual variance not explained by these bigram and trigram frequencies, particularly after extensive training (Knowlton & Squire, 1996; Meulmans & Van der Linden, 1997). Johnstone and Shanks (1999) showed that this extra variance in the Meulmans and Van der Linden study could be accounted for by assuming people learned the positional constraints of the trigrams; or, what amounts to the same thing for the structures they were dealing with, tetragrams.

In summary, with continued exposure to exemplars of an artificial grammar, people learn successively higher orders of redundancy. Because connectionist networks are ideal devices

for modelling the extraction of statistical redundancy, these models have been used in the past to successfully account for various aspects of artificial grammar learning. We will now turn to consider two key models in this area.

3. The connectionist models

3.1. The trouble with parameters

Many cognitive theories and computer simulations take parameters that govern aspects of their behaviour. A model with many parameters, or with parameters that have a large effect upon performance, is often criticised. However, models do not need to be rejected just because of their parameter dependence. If a parameter has some obvious meaning within the context of some explanation, or perhaps some physiological correlate, we might have an *a priori* reason to set it at some specific value. With many parameters in many computational models in psychology, however, there is no directly measurable physiological or cognitive correlate and so there is no reason why its value should be committed before hand. This is the case for, say, the parameters governing back propagation.

Simply searching for those parameter values that produce the best (i.e., most human-like) performance, might be considered as producing an existence proof that the system *can* reproduce human behaviour. This provides only *post hoc* justification for a particular choice of parameters. Demonstrating equivalent performance to people on new stimuli with the same parameter values would lend non *post hoc* support to both that system and those parameter settings (e.g., Kruschke & Johansen, 1999; Servan-Schreiber & Anderson, 1990). Further, one might be able to justify both the system's and people's arrival at those settings in terms of evolution if the parameter values can be seen as providing an optimal solution to ecological problems.

Often in the past, researchers have tried to fit a single value for each parameter based on the mean behaviour of a group of people (e.g., Dienes, 1992). It may be more natural to presume that variability in one or more parameters represents the natural variability found between different people. The stochastic behaviour of a probabilistic model could be seen as representing within person variability, whilst the different parameter settings could be seen as representing between person differences (Nosofsky, Palmeri, & McKonley, 1994). To examine such a claim the behaviour of the model over a range of parameter settings should be explored. (In real neural networks, there must be evolutionarily-tuned additional parameters that specifically constrain the range of the other model parameter values that people vary over.) If an explanatory claim of some parameter-laden system is to be made, then the parameters of that system should be examined rather than avoided.

We will investigate our models across a broad range of parameter settings to get a qualitative description of its behaviour rather than just individual existence proofs. In this way, each model can be characterised and even though several models might all be able to produce human-like performance with some set of parameters, the model for which human-like performance is more characteristic would be preferred. The argument can be construed as a Bayesian one: we will prefer the model that has the highest probability density (over the space of measured dependent variables) for the behaviour that matches human behaviour (Bishop, 1996).

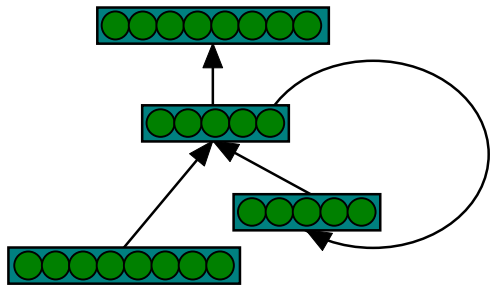

Fig. 1. The architecture of a simple recurrent network.

3.2. Simple recurrent network

3.2.1. The model

The simple recurrent network (Elman, 1990) is a variant of the traditional three layer feed-forward artificial neural network. The difference is the addition of extra input units, called the context layer, used internally by the network to provide a dynamic memory for previous hidden layer activations. Every time the network is used, the hidden unit activations are copied onto these context units and preserved for use with the next input. This architecture is shown in Fig. 1. The recurrent connections, that copy hidden unit activation into the context layer, generally contain no weights and are not trained. This means that, apart from the copy procedure, an SRN is trained and run as a feed-forward network. Formally, the SRN is a recursive function operating on, and returning, a variable length sequence of vectors. This function is as universal as that of a basic three-layer network but has the operational advantage of input length flexibility.

Cleeremans (1993) first showed how the SRN could learn finite state grammars, and how it provided a good computational model of the implicit learning of finite state grammars in a sequential reaction time paradigm. Dienes (1993) extended this work to the artificial grammar learning paradigm introduced by Reber (1967). In the application of the SRN in this paper, the network is used to predict the next letter of any sequence. In the memorisation phase letters from each exemplar are input to the network one at a time from left to right. With each input, activations are fed forward through the network. The output is compared to the next letter of that sequence, with the difference between "predicted" successor and actual successor used to calculate the error for back propagation training (Rumelhart, Hinton, & Williams, 1986).

In this paper, language exemplars have a dummy start and end symbol to mark their boundaries, and an orthogonal representation is used for each letter in both the input and output layers. Specifically there is one unit for every possible letter (and for the start and end symbol) in the training and test languages and the input and desired output vectors use activations of 0.9 (or 1.0) for the unit representing the current letter and 0.1 (or 0.0) for all other units.

In the classification phase, back propagation is left on.[1] Each test exemplar is fed through the network in the same way as in the memorisation phase. The predicted and actual output activations are stored for each letter position and joined end to end to make one vector for the whole exemplar of actual output activations and one vector for the whole exemplar of

predicted activations (a procedure used previously for modelling of artificial grammar learning by Altmann & Dienes, 1999; Berry & Dienes, 1993; Dienes, Altmann, & Gao, 1999; Kinder & Shanks, 2001). The Pearson correlation between the two vectors is used to determine how close the SRN's predictions are to the actual stimuli: It is assumed that the better the SRN's predictions the more likely the exemplar is to be grammatical. Given the output correlation, c, the probability of classifying an exemplar as grammatical is given by the formula (1). (Note that individual people also classify probabilistically in artificial grammar learning tasks; Dienes, Kurz, Bernhaupt, & Perner, 1997.) A classification threshold, T, is chosen so as to classify approximately half of the exemplars as grammatical and half of them as non-grammatical. The average output correlation across the whole test set was used to set the value of T.

$$p(\text{“}g\text{”}) = \frac{1}{1 + e^{T-c}} \tag{1}$$

where T, is a classification threshold and c, is the output correlation.

The SRN is proposed as a computational and psychological theory of grammar learning. What can we say about the cognitive process that such an SRN implements? There are a number of factors to consider, all relating to design decisions made in implementing the system and adapting it for grammar learning. The input and output representations, the simulation regime and the learning algorithm are considered in turn.

The system assumes pre-processed and independent letter representations. This is not unlike most other theories of grammar learning, apart from the two-layer connectionist simulation described by Dienes (1992). The latter used separate representations for letters in different positions in the stimuli. So, unlike that simulation, an SRN makes no *a priori* distinction between a letter at the beginning of an exemplar and the same letter half way through it. The network does have some sense of letter order however, as it parses its stimuli one letter at a time from left to right and this is distinct from many other models (such as Competitive Chunking, or the Memory Array/Exemplar models; see Berry & Dienes, 1993) which effectively treat exemplars as occurring at single points in time, and so the SRN offers a potential scope advantage over these other models in terms of accounting for sequence response/prediction tasks (Cleeremans, 1993; Cleeremans & McClelland, 1991).

As used in this paper, the SRN learns “incidentally” in the sense that back propagation is not turned off between the memorisation and test phases. This might be seen as analogous to an automatic process that proceeds despite one's intention. Whether the SRN models a perception or a memorisation process (if we can make a distinction between the two) is a moot point (cf. Kinder & Shanks, 2001). It can only learn sequential structures so it cannot be seen as a general model of perception. On the other hand, the way in which the SRN could presumably reconstruct partially degraded grammatical stimuli suggests it as a model of perceptual fluency. The SRN could be seen as attempting to memorise sequences of stimuli. On the one hand, its learning is unlike rote learning because it does not in general attempt to create a representation of each separate sequence but rather capture the structure of a set of sequences, as discussed below. On the other hand, in the current implementation the SRN does this without reference to any pre-existing related knowledge, and in this way it does engage in a process similar to rote-learning (contrast Altmann, in press, in which an SRN makes use of pre-existing knowledge).

The SRN has been previously characterised by the work of Elman (1990) and Cleeremans, Servan-Schreiber, and McClelland (1989), who highlighted two main issues. Firstly, with learning, the SRN acquires a gradual sensitivity to an ever-increasing number of elements of the preceding sequence, starting with one element and then two elements, etc. Thus, to some extent the knowledge acquired by the SRN could be characterised as fragmentary. The SRN becomes sensitive to a number of small and overlapping sequence fragments. This is distinct from memory array models in which exemplars are memorised as whole traces. As will be seen the SRN is remarkably similar to the Competitive Chunking (CC) model in this respect as the CC model also learns fragmentary information. Elman illustrated how the SRN could behave as if it had formed "chunks" of frequently occurring sequence fragments. He exposed an SRN to a long sequence of phonemes corresponding to words in the English language; the SRN's task was to predict the next phoneme that would occur. After sufficient exposure, the SRN could predict with little error within a word, but with considerable error between words. Roughly speaking, the SRN had learnt chunks of phonemes that corresponded to words, but those "chunks" were not represented locally within the SRN; instead, the knowledge was embedded in a distributed way amongst all its weights.

Secondly, an SRN actually learns the structure of a sequence rather than just the sequence itself. So, Elman (1990), using natural language stimuli, found that the SRN allocated distinct areas of its hidden unit activation space to represent different grammatical categories, such as nouns, verbs, and also finer distinctions. Cleeremans et al. (1989) using a finite state grammar found that an SRN's hidden units came to encode the states of a simple finite state grammar. Their network was trained to predict the next letter in sequences generated by the grammar and a cluster analysis of the hidden unit activations revealed distinct clusters for each of the grammar's states (or non-terminals). As with Elman's work, the suggestion is that the network has abstracted certain structural regularities from the stimuli. In general, however, the SRN does not form structurally explicit representations of the finite state grammar used to generate the stimuli; instead the structure of the grammar is implicit in the weights as a whole (Dienes & Perner, 1996).

3.2.2. Parameter ranges

The aim of computer simulation in this paper is not just to find those parameter settings that can achieve some behavioural performance but also to characterise a model across a range of possible parameters settings. Since the SRN has a number of parameters, and they can each take a large range of possible values, it seems pragmatic to define a range within which simulation is performed. This range has to be large enough to capture most of the SRN's possible behaviours. Table 1 shows the range of parameter values used in all the simulations reported in this paper. The ranges seemed to offer a reasonable compromise between practical feasibility and behavioural coverage. Most of the reported parameter settings from existing grammar learning and connectionist simulation work are included in this range.

The iteration parameter is potentially different from the others because it does have some physical correlate, but not one that we can match to the human experiment very easily. The longer an exemplar is exposed, the more time there is to process it, and there must be a relationship between exposure duration and the number of iterations. However, it is not clear what this relationship is, or even that it is linear. For the purposes of this paper, a single standard exposure

Table 1
The standard range of parameter and regime settings for the SRN

Setting	Description	Standard range
Parameters		
l	Learning rate: the rate at which weights change under back prop	0.1, 0.3, 0.5, 0.7, 0.9
m	Momentum: the degree to which weights continue to change in the same direction	0.1, 0.3, 0.5, 0.7, 0.9
a	Architecture: the number of hidden units	5, 7, 9, 11
i	Iterations: the number of times an exemplar is presented before progressing on to the next one	1, 2, 3, 5, 7, 9
w	Initialisation weights for SRNs	Randomly selected from the range 0.1 to 0.9
Regime		
e	Epochs: the number of cycles through the training set	1, 2, 3, 4, 5
n	The number of SRN's run with each combination of *l*, *m*, *a*, and *i*. Each of these *n* different SRNs are distinguished by having randomly different *w*	20

time for both the human experiments was used, 750 ms per exemplar letter. The largest value of the iteration parameter was 9, corresponding to a processing time of about 80 ms per letter per iteration.

Although a learning rate of 0.9 might seem large, it was one of the best fit values found by Dienes et al. (1999) in simulating an artificial grammar learning experiment (in one case, Dienes et al. found a best fit learning rate of 1.1). Kinder and Shanks (2001) also explored learning rates up to 0.9 in simulating data from artificial grammar learning experiments.

3.3. Competitive chunking network

3.3.1. The model

Servan-Schreiber and Anderson (1990) developed their competitive chunking model to account for people's ability to detect regularities in the environment incidentally. Their specific test case was artificial grammar learning, but the model was seen as applicable to incidental learning generally. They postulated that often the learning mechanism involved is chunking and the resulting knowledge is a hierarchical network of chunks.

The competitive chunking model of Servan-Schreiber and Anderson (1990) can be interpreted as a connectionist network.[2] This network is feed forward and has a number of intermediate layers in which each unit represents some pre-determined fragment, or chunk, of an input stimulus. Chunks are hierarchical so successive intermediate layers represent increasingly complex groupings of the chunks represented in earlier layers. This is illustrated in Fig. 2 for a single stimulus "MTVR."

The competitive chunking model perceives a stimulus by successively chunking together the basic components of that stimulus until a single chunk represents it. So, using brackets to denote a chunk, the exemplar "MTVR" might be perceived as first "(MT)VR," then "(MT)(VR)" and finally "((MT)(VR))," In the competitive chunking network, chunking a stimulus component is represented by the activation of that chunk's unit.

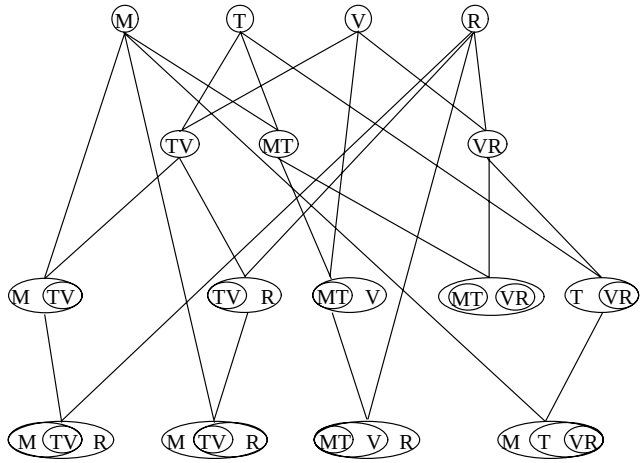

Fig. 2. The competitive chunking network architecture for processing the stimulus 'MTVR'.

Perception is understood as a wave of activation passing through the network from basic chunks in the first layer, to complex chunks in later layers. This is illustrated in Fig. 3 for the pattern of chunking used above on the "MTVR" stimulus.

Once a stimulus is fully chunked it is said to be maximally familiar, or memorised. So, activation of a unit in the final layer of the network, or in fact any unit that represents the stimulus as a single chunk, signifies memorisation of that stimulus. However, when the network is used

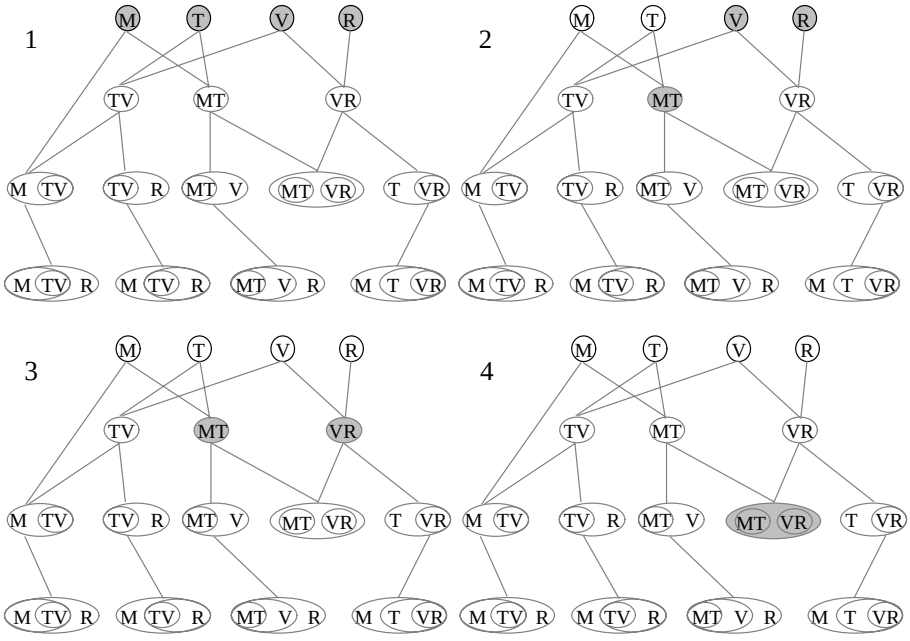

Fig. 3. A possible flow of activation through the competitive chunking network to perceive the stimulus 'MTVR'.

for classification tasks the perception process can fail to activate a stimulus chunk causing the flow of activation to stop at some non-final unit. In this case, the familiarity of the stimulus is determined by the number of chunks into which that stimulus has been perceived, or in other words, by the number of active units. The fewer active units the more familiar the stimulus.

According to Servan-Schreiber and Anderson (1990), competitive chunking can operate in two modes, memorisation and classification. The difference between these two modes is that in memorisation the network can grow, starting with just an input layer growing intermediate units for chunks as they are needed, whilst in classification the network's intermediate architecture is fixed.

Input to the competitive chunking network is a representation of the perceptual elements of a stimulus along with their relative positions. So, "MTVR" would be encoded as an "M" in position one, a "T" in position two, a "V" in position three and an "R" in position four. Whilst this might suggest that the system has no way of knowing that an "M" in first position is in any way related to an "M" in third position (as in the connectionist model of Dienes, 1992), this information is in fact encoded implicitly in the connections from input units to their relevant chunks.

When a new stimulus is presented to the network the appropriate letter-position input units are activated and this activation is fed forward through the network. Activation in all units is binary, in that a unit is either active or inactive. However, units also have both a support and strength that are used to determine which units become active. Learning is mediated by incrementing a unit's strength whenever that unit has been activated, as described below.

A unit's support is the average of the strengths of the active inputs to that unit, as defined in formula (2)

$$\text{Support} = \frac{\sum(S_i \times A_i)}{n} \tag{2}$$

where S_i is the strength of input unit i, A_i is the activation of input unit i, and n is the number of input units.

Every unit with some support can become potentially active. Potential activity is a stochastic function of a unit's support and the competition parameter c ($0 < c$), the probability of which is given in formula (3)

$$p(\text{potential}) = \frac{1 - \mathrm{e}^{-c\,\text{support}}}{1 + \mathrm{e}^{-c\,\text{support}}} \tag{3}$$

Of all the potentially active units the one with the greatest strength is made active. We can interpret this by equating a unit's strength with response time so that the unit with the greatest strength becomes active first. As soon as a unit is activated its input units loose their activation and can no longer contribute to potential activity in other units. So, for the "MTVR" stimulus, as soon as the unit representing "(MT)" is activated the input units "M" and "T" become inactive. This prevents the network from activating units that represent overlapping chunks, such as "(TV)."[3] This process of activating the strongest chunk and deactivating its inputs continues until there are no more potentially active units. At this point, the wave of activation can then move on to the next layer.

Whenever a unit is activated its strength is incremented by a single time-decaying value. A unit's strength is calculated as the sum of each of these decaying increments. The decay is

governed by the decay parameter, d ($0 < d < 1$), and is such that once created, a chunk's strength can never actually reach zero.

There are two reasons why the flow of activation through a network may fail at some point prior to a final unit. These are that, firstly, since the network grows on the basis of experience, the unit for some stimulus chunk might not actually exist; and, secondly, that the process of unit activation is stochastic. In memorisation modes, when either of these situations occurs, and the active unit does not represent the whole stimulus, the network grows a new unit. Candidates for new units are drawn from the set of nodes representing all possible pairings of currently active units. From this set, the unit with the highest support, given the activation pattern when activation flow stops, is added to the network and activated. There is no limit to the number of new units that can be created.

In classification mode whenever activation flow stops the chunking process stops. The familiarity of a stimulus is then determined by the number of active units using formula (4)

$$\text{Familiarity} = \mathrm{e}^{1-\text{nactive}} \tag{4}$$

Competitive chunking makes classifications on the basis of familiarity as in formula (5), which defines the probability of calling a test string grammatical.

$$p(\text{“}g\text{”}) = \frac{1}{1 + \mathrm{e}^{T-\text{nactive}}} \tag{5}$$

where T, is a classification threshold.

This formula is the same as that used by the SRN (1) only based on the number of active units rather than output correlation. Servan-Schreiber and Anderson also used only the last 20 stimuli as a running average upon which to base their classification threshold, rather than the whole test set.

As with the SRN, the types of cognitive processes and representations that the CCN proposes should be considered. Unlike the SRN, however, there is probably less to say, as the CCN is a more explicit model. Input to the CCN is in terms of an ordered sequence of letters. Each letter is represented with an independent symbol, much like the SRN. Unlike the SRN, however, the CCN processes stimuli as a whole, making no commitment to parsing an exemplar from left to right. Instead an exemplar needs to have been completely processed in an input buffer before chunking can proceed.

The CCN is designed to model two types of behaviour, memorisation and classification. In fact, it suggests that both abilities are mediated by the same perceptual process, chunking. Familiarity is a recognition measure which like classification probability is based upon the number of active units.

In this description memorisation is an intentional rather than incidental addition to the perceptual process and is enabled by allowing the creation of new chunk traces in memory. So, for the CCN there is a distinction between learning and classification and one might be tempted to argue that this doesn't really account for incidental learning effects; for example those found by Gordon and Holyoak (1983). However, the process of memorisation may indeed be incidental to normal perception but subjects can distinguish processing that occurred in different contexts (as found by Dienes, Altmann, Kwan, & Goode, 1995). A classification scenario may present a context which can be used by subjects to separate the processing that

went on in a previous context. (This is simplistic; the way knowledge in different contexts may be kept separate is an interesting modelling problem in itself; cf. Broeder & Plunkett, 1994.) If that were the case, it would seem that learning had been "switched off" in the classification phase, as assumed by Servan-Schreiber and Anderson (1990).

Buchner (1994) also attempted to show a dissociation between the two types of behaviour that CCN claims are mediated by one process—memory and classification. Buchner showed that after training, subjects were faster to identify perceptually degraded strings if they were grammatical rather than nongrammatical. This perceptual familiarity was shown to be related to recognition judgement, in that "old" responses were associated with faster identifications than "new" responses. There was no significant relationship between speed and grammaticality judgements, but Buchner did not report what power the study had to detect the size of effect found for recognition judgements, so the findings do not compromise the CCN model.

We should also consider the type of knowledge that the CCN predicts chunking to produce. In Dienes and Perner's (1996, 1999) terms, everything in chunk memory is property-structure explicit, that is, compositional. The chunk "(V(RX))" represents the fact that "V" is followed by an "(RX)," where "V" is a basic perceptual unit (a representational building block) and "(RX)" is itself a property-structure explicit representation of "R" followed by "X." The strength of these representations is also a separable sub-component.

The process of chunking means that the CCN also has fragmentary knowledge, which it builds up over time, becoming gradually sensitive to longer length sequences of letters. And, like the SRN, it learns by sharing commonly occurring low level chunks between a number of high level representations: i.e., it learns by removing sequential redundancy. However, unlike the SRN this knowledge cannot be abstract. Every representation in chunk memory can be separated into its basic perceptual components.

So, for example, imagine a finite state grammar with a state, S1, that can be reached via the sequences "AB," "MB" and "VT," and that allows "X" or "M" to follow. An SRN might learn to use a single pattern across the hidden units to represent this state, and any of the three preceding sequences would lead to its high activation. The connections from this pattern would then be weighted strongly in favour of the output units "X" and "M." The same knowledge in a CCN would have to be encoded in the form of six meta-level chunks: "((AB)X)," "((AB)M)," "((MB)X)," "((MB)M)," "((VT)X)," "((VT)M)," with no link to suggest that these might all be examples of the same component of structure.

3.3.2. Parameter ranges

The range of parameter values for the simulations of the CCN reported in this paper are listed in Table 2.

Servan-Schreiber and Anderson (1990) specified that the decay parameter should vary in the range $0 < d < 1$; hence, we have spread the parameter values out evenly over this range. But why should this particular range be used? The decay parameter is used in determining a chunk's strength, which is the sum of its successive individually decaying strengthenings:

$$\text{Strength} = \sum_i T_i^{-d}$$

where T_i is the time elapsed since the ith strengthening. This is the same formula as is used in Anderson's ACT model (e.g., Anderson, 1983, where the parameter is restricted to the same

Table 2
The standard range of parameter and regime settings for the CCN

Setting	Description	Standard range
Parameters		
c	Competition: governs the degree with which increased support increases the likelihood of chunk use	0.1, 0.3, 0.5, 0.7, 0.9
d	Decay: rate at which a chunk's strength decays over time	0.1, 0.3, 0.5, 0.7, 0.9
i	Iterations: the number of times an exemplar is fully perceived (so that nactive = 1) before progressing on to the next one	1, 2, 3, 5, 7, 9
Regime		
e	Epochs: the number of cycles through the training set	1, 2, 3, 4, 5
n	The number of CCN's used with each combination of the above parameter values	20

range, p. 174). The reason for the limits (between 0 and 1) is that they result in total strength growing as an approximate power law (Anderson, 1983, p. 182). The power law of practice was hailed by Newell (1990) as a regular, robust and ubiquitous law of practice; a law that Newell himself explained in terms of chunking. Hence, there are strong *a priori* reasons for the limits given for d.

The parameter values for c have been spread over the same range. The only absolute constraint is that $c > 0$; however, Servan-Schreiber and Anderson (1990) explicitly considered c values only up to 1. The c parameter acts as a type of learning rate; the higher c is, the more available already formed chunks are for use, and the faster learning is. Servan-Schreiber and Anderson found a good fit for their data with $c = 0.5$; Dienes (1993) found $c = 0.5$ produced too much learning (model performance around 80% when people performed around 65%).

There are four differences between this CCN implementation and the competitive chunking model of Servan-Schreiber and Anderson (1990). They are listed below.

1. *A chunk is activated only once in a stimulus.* The network described here allows a chunk to compete for use in representation of a stimulus only once in that stimulus. This means that stimuli of the form "ABXAB' might have to be processed with two separate units to represent "AB." This fact does not affect any of the experimental manipulations considered in this paper.
2. *No length three chunks or concatenation of repetitions.* For the straightforward reason of implementational simplicity the CCN was prevented from being able to create chunks greater than length two. Servan-Schreiber and Anderson's (1990) model could chunk sequences of length three and repeated sequences of any length. Although such a facility might be useful to the CCN it does not affect any of the analyses or stimuli used here.
3. *No familiarity running average.* Both the CCN and SRN make classifications by comparing some measure of exemplar grammaticality with a threshold. Our implementations of the SRN and CCN use a threshold determined over the whole training set, so as to help ensure a fifty-fifty classification bias. Servan-Schreiber and Anderson's (1990) CCN, on the other hand, bases it on a running average of the last twenty exemplars. Whilst the

latter approach doesn't require the model to see every test exemplar before classifying the first one, it effectively adds an extra parameter—running average size—which would have to be included in the model characterisations. Bias rates, along with a system's ability to model them, are interesting areas of investigation, but are not a concern of this paper.

4. *No memorisation criterion.* the CCN was not trained to a preset memorisation criterion during training for two reasons. First, the CCN was not seen as providing a model of recall (as it is, the CCN operates on a presented stimulus, it does not generate stimuli). Further, comparison between the CCN and SRN is easier, the more similar the training regime.

4. Learning basic associations

In this section how CCN and SRN simulations treat letter pairs when learning artificial grammars is examined. The letter pair (bigram) corresponds to one of the fundamental levels of learning, that of basic association, and is a well-researched area in both the sequence and concept learning domains. Examples of existing work include that of Cleeremans (1993), Gluck and Bower (1988), and Shanks, Charles, Darby, and Azmi (1998).

For both the SRN and the CCN, the letter pair is a crucial object of the learning mechanisms. The CCN uses explicit chunks of letter pairs as building blocks for higher-level structure and the SRN initially learns to predict which letter will follow any current one. So, differences in the way letter pairs are learned or used might lead to interesting differences in behaviour.

The work presented here is motivated by two initial observations.

(i) The CCN learns bigrams as single chunks with an associated strength. In contrast, the SRN is trained to predict the next letter in a sequence. For the SRN this can lead to *prediction conflicts* where a letter, "A" say, is legally followed by a number of other letters, say "B" and "C." The more equi-probable letters that are permissible following an input ("A"), the lower is the activation of any of the corresponding output units after the SRN has been asymptotically trained. For the CCN, there simply is no conflict as "AB" and "AC" are represented by two independent chunks.

(ii) The SRN's training is output driven (sometimes called discriminative training). That is, for a certain input the SRN learns some desired output. If at some point in training the desired output changes, the SRN will simply forget, or overwrite, its previously learned output. This effect is often known as retroactive interference or, in a connectionist context, catastrophic forgetting (McCloskey & Cohen, 1989). The CCN, on the other hand, builds up a representation of its input structure that, though decaying gradually over time, is never overwritten by new input.

These two observations are related, but different—forgetting is perhaps just one aspect of the predictive way in which the SRN learns bigrams.

The experiments described below exploit these two observations to separate the characteristic behaviours of the two models. This section will start with a more detailed analysis of bigram

learning in the SRN and CCN followed by designs for two experiments. Each experiment is then reported in turn with the results discussed in the final section of this paper.

4.1. Factors affecting bigram learning

Initially, the SRN and CCN are analysed with respect to their bigram representational abilities and the effect such representations might have on classification. Then the effects of exposure on bigram learning are considered.

4.1.1. Bigram representations and classification

The SRN is trained to predict the letter that follows some current letter. It learns to make predictions on the basis of an input letter but with further training (as described by Cleeremans, 1993) learns to take longer preceding letter sequences into account. It is the initial moment of simple bigram sensitivity that we are interested in. In that moment, the SRN's bigram knowledge can be expressed in terms of a number of contingencies denoting which letters can follow which. The output activations of the SRN then reflect the conditional probabilities of every letter given any current one: $p(L_{i+1}/L_i)$, the probability of L_{i+1} given L_i where L_i and L_{i+1} are successive letters in a sequence. Whether the SRN's knowledge actually reaches this state, before acquiring higher-order sensitivity, and what effect a higher-order sensitivity will have on bigram predictions is an open question. However, for now we will assume that there is some point in training where this learning is approached. If a letter (say A) can be legally followed by many other equi-probable letters (e.g., if AA, AB, AC, AD and AE are all grammatical), then the output units representing the possible successors of A will have lower activations than if A were to be followed by only a few letters.

In contrast, the CCN does not learn conditional probabilities. The CCN represents its bigrams as a single chunk, with some associated strength. This strength increases with bigram occurrence, and, though it decays gradually over time and is not guaranteed to increment with every occurrence of that bigram, correlates with bigram frequency: $F(L_i, L_{i+1})$, the frequency of L_i and L_{i+1} where L_i and L_{i+1} are successive letters in the sequence. The notable exception to this occurrence-based strength incrementation, comes from the fact that a CCN can only form one bigram chunk out of every trigram it encounters, so knowledge of a very frequent bigram could block learning a less frequent overlapping bigram (this would allow it to apparently learn conditional probabilities in stimuli like those used by Aslin, Saffran, & Newport, 1998).

However, with the CCN chunk strength is not used in classification directly. So, the CCN might be expected to be more sensitive to bigram violations (i.e., the appearance of bigrams in the test phase that had never occurred in training) than relative differences in bigram frequency (cf. the sensitivity of people to "chunk novelty," Meulmans & Van der Linden, 1997). Like the SRN, however, the CCN's knowledge of higher order structure would be expected to increase with training, although this is unlikely to affect bigram chunking.

4.1.2. Exposure

From the above analysis it follows that the relative frequency and arrangement of bigrams in a training set should have an effect on how these bigrams are learned by the two models. Consider first relative frequency. If one bigram, "LM," occurs twice as often as another, "XY,"

in a training set, then what effect does that have on judgements of grammaticality by the two models? The CCN will favour an "LM" chunk over "XY" in rough proportion to its relative frequency. However, any difference in chunk strength is unlikely to affect the CCN's final classification unless a conflict is created in which the use of one chunk prevents the use of another. Even with a forced choice test between two strings, each containing one of the pairs "LM" or "XY," the chunks' strengths will not affect overall stimulus familiarity, and the strings will be classified on the basis of some other factor. Only if the bigrams were overlapping (e.g., the bigrams "LM" and "MY" overlap in the string "LMY") might different bigram frequencies influence classification choices.

For the SRN, frequency is also only important under certain conditions. Once a contingency is learned, then so long as there is no conflict or forgetting, increased learning will have no effect upon the accuracy of the SRN's output prediction. If both "LM" and "XY" have been seen enough, then the fact that "LM" occurs twice as often as "XY" is not important, "L" will predict "M" and "X" will predict "Y." This is not the case, however, with prediction conflicts where if, say, "AB" occurs twice as often as "AC," the SRN's predictions will reflect the bigram conditional probabilities. So, "A" will more strongly predict "B" than "C."

In summary, the CCN and SRN are sensitive to relative frequency in different ways, depending upon the type of bigram. Thus, one way of distinguishing the predictions of the models would be to manipulate the frequency with which different bigrams occur.

Another factor that affects performance differently in the two models is whether the frequency of bigrams changes as learning proceeds. It is well known that the standard back propagation network is susceptible to catastrophic forgetting; i.e., it is very sensitive to retroactive interference. Because the SRN learns by back propagation, it should be possible to produce substantial retroactive interference. For example, consider an SRN learning stimuli containing a prediction conflict. If one bigram "AB" only occurs in the first half of training, and another bigram "AC" only occurs in the second half of training, then learning "AC" should cause the SRN to unlearn "AB." On the other hand, the CCN does not learn by error correction, and the learning of "AC" will have no consequence for its prior knowledge of the bigram "AB."

4.2. Experimental manipulations

4.2.1. Relative frequency prediction conflict

The first manipulation aims to exploit the sensitivity of SRNs to prediction conflicts by varying the frequency of the prediction conflict bigrams. For this, "AB" is presented twice as often as "AC" in training. The SRN would be expected to favour "AB" over "AC" whilst the CCN should favour neither bigram.

4.2.2. Retroactive interference (distributed prediction conflict)

The second manipulation exploits the SRNs sensitivity to retroactive interference. The learning phase contains an equal frequency but uneven distribution of conflict bigrams: "AB" occurs in only the first half of the training phase, and "AC" occurs in only the second half of the training phase. In testing, "AC" will be favoured by an SRN over "AB," as "AB" will have been forgotten. For the CCN, on the other hand, "AC" will only be favoured over "AB" to the extent that the strength of "AB" has decayed. The amount of decay can be measured by including a control

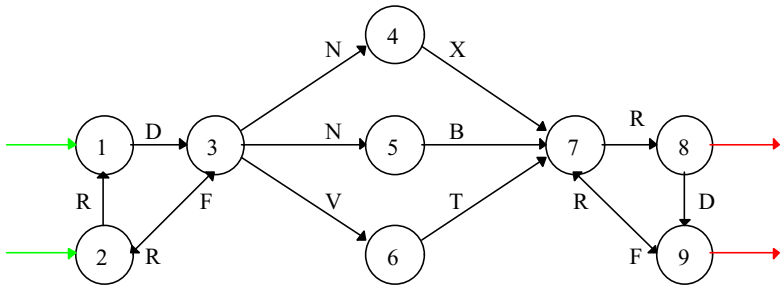

Fig. 4. Training grammar for the relative frequency prediction conflict experiment.

condition in which a non-conflict bigram is substituted for the conflict bigram; for example, "DC" instead of "AC." The difference in the tendency to endorse "DC" over "AB" provides a baseline measure of decay. In the SRN, the tendency to endorse "AC" rather than "AB" should be greater than the baseline measure of decay; in the CCN the tendencies should be equal.

In order to ensure that the same cognitive processes are being investigated in the experiments reported here as in the standard artificial grammar learning paradigm, conflict bigrams were embedded within longer strings.

4.2.3. Relative frequency prediction conflict

For this experiment the training language must satisfy the simple constraint of containing twice the number of one type of conflict bigram rather than the other. The grammar (Fig. 4) satisfies this constraint.

The training grammar for this test is pictured in Fig. 4 and the training strings are shown in Table 3. The crucial bigrams are "DN," "FN," "DV" and "FV" with "DN" occurring twice as often as "DV," and "FN" occurring twice as often as "FV," "DN" and "FN" are the high frequency conflict bigrams and "DV" and "FV" are the low frequency conflict bigrams.

The "conflict test set" was completely grammatical, and contained an equal frequency of all the crucial bigrams. Any favour for the high frequency conflict bigrams should tend to produce grammatical responses to exemplars in which they are contained, and non-grammatical responses to exemplars containing the lower frequency pair. The "control test set" presents a

Table 3
The training language for the relative frequency prediction conflict experiment, highlighting low frequency bigram exemplars

Training set		
FRFNXF	DRFNXR	**DVTF**
DNBR	**FRFVTF**	DRFNBF
FNBR	FNXR	FRFNBF
DVTR	RDNXF	DNXF
DRFNBR	**RDVTF**	**DRFVTF**
FVTR	DRFNXF	RDNBF
DNBF	**DRFVTR**	DNXR

Table 4
Conflict and control test sets for the relative frequency prediction conflict experiment

Conflict		Control	
DVTRD	**DVTFRF**	**DVTRR**	DVTFRF
DNBFRR	FNXFRR	DNBFRR	**FNXFRD**
RDNXRD	**FVTFRF**	**RFNXRD**	FVTFRF
FVTF	FRFNXR	**FVTRFR**	FRFNXR
FNXF	**FVTF**	**FNXD**	FVTF
RDVTRD	**RDVTRD**	FDVTRD	RDVTRD
DVTRD	RDNBRD	**DVNRD**	**FDNBRD**
FVTFRR	FNBRD	**FNTFRR**	FNBRD
FNBF	RDNXR	FNBF	**RDXR**
FNBFRF	RDNBR	**FRBFRF**	RDNBR
DVTFRR	**RDVTR**	**DVTFFR**	RDVTR
DNBFRF	FRFNBR	DNBFRF	**FRFNRR**
FNXFRF	FNXRD	FNXFRF	FNXRD
FRFVTR	FNBFRR	**DVFVTR**	**FNBRR**
FVTFRR	DNBRD	**FVNFRR**	DNBRD
DNXFRR	DNXRD	**DNXRFR**	**DNXRF**
DVTFRR	**FRFVTR**	**DVTFRN**	FRFVTR
DVTFRF	**FVTRD**	**DVTFRD**	FVTRD
FVTFRF	**RDVTR**	FVTFRF	**RDVTD**
DNXFRF	**FVTRD**	DNXFRF	FVTRD

Bold text highlights the conflict low frequency bigrams and the control bigram violations.

traditional test, containing equal numbers of grammatical and non-grammatical exemplars. For this, the non-grammatical exemplars are simple copies of the grammatical ones but with added bigram violations. General bigram sensitivity should be reflected in grammatical responses to grammatical exemplars.

In the scoring below for the experimental test set, the "DV" and "FV" (low frequency conflict bigrams) test exemplars were treated as grammatical, and the "DN" and "FN" (high frequency conflict bigrams) test exemplars as nongrammatical. Assume that there is adequate learning shown on the control test set (by the person or computational model); that is, the relationships between immediately successive letters have been encoded in some way. Then if the coding (by the person or model) is sensitive to prediction-conflict, performance on the conflict test set should be significantly below 50%; if the coding is insensitive to prediction conflict, performance should be 50%, other factors being equal.

Both test sets are contained in Table 4. A between-subjects experimental design was used for people, with two groups. Both groups were exposed to the same training set, but were tested differently: One group was tested with the conflict test set, and the other with the control test set.

4.2.3.1. Computer simulations. As explained in the introduction, it is useful to know the general types of behaviour that each model can produce. So, instead of finding an example implementation that satisfies some behavioural criterion, the models are examined as a class, across many parameter and exposure settings. Both the CCN and SRN were tested on this experiment

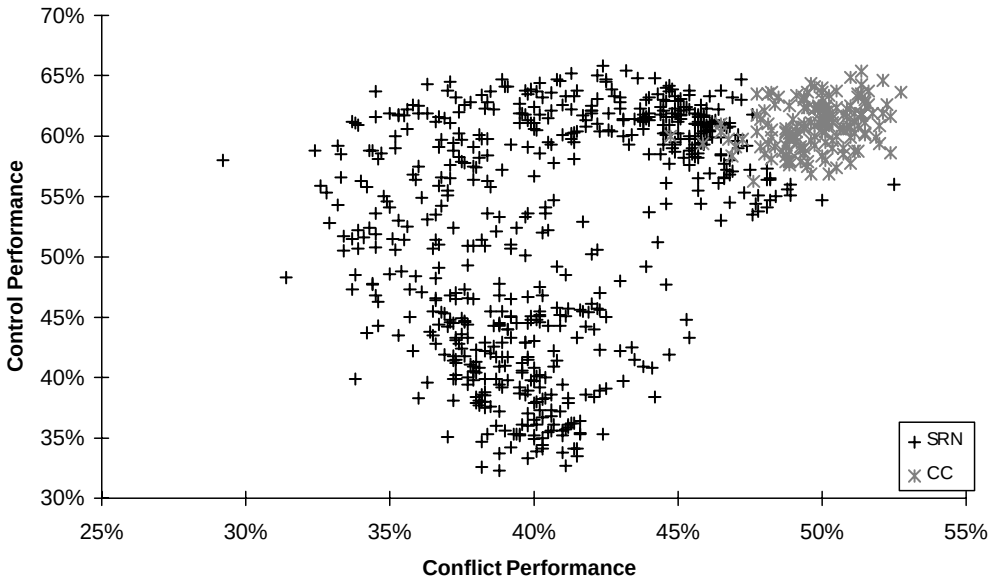

Fig. 5. SRN and CCN performance on the relative frequency prediction conflict experiment. The plot contrasts performance on the control and conflict conditions.

across the range of parameters given in Sections 3.2.2 and 3.3.2 above. This meant running six hundred SRNs and one hundred and fifty CCNs for each epoch's exposure—remembering that the CCN has one less free parameter. Only the five epoch training results are displayed here (there is little overall difference between the three and five epoch performances). The human experiment, described below, also used five epochs.

The two conditions, control and conflict, are compared for every simulation parameter setting, and the scatter plot above (Fig. 5), shows performance on the control against conflict conditions. As we can see, the CCN remains insensitive to prediction conflicts as bigram knowledge (measured by the control condition) increases; exactly as predicted. In fact, CCN performance on the conflict condition varies between only 52.75% and 44.75% across the whole parameter set.

The SRN, on the other hand, remains consistently sensitive to prediction conflict frequency; again in the way that had been predicted. SRN performance on the conflict condition ranges between 52.50% and 29.20%, showing a strong preference for higher conditional-probability bigrams. But is this difference enough to separate the two simulations? The SRN's behaviour on the conflict condition completely contains the CCN's performance, and even taking the control condition into account there is still a reasonable range of results that both simulations could produce.

There are two levels at which we can distinguish these simulations, one descriptive and the other quantitative. At a descriptive level, a substantial part of the SRN's performance is affected by the prediction conflict manipulation, and the system can be characterised as sensitive to it. The overlap in performance between the SRN and CCN is contained in a section of the SRN's behaviour that is not obviously affected by the manipulation and is, in that sense

uncharacteristic. For the CCN, its overlapping performance is still characteristic of its general behaviour under prediction conflicts. Were people's performance on this test to lie in that overlap region we would still favour the CCN over the SRN for a number of reasons, even though both systems could plausibly simulate people in this case.

One reason for supporting the CCN would be Popperian (Popper, 1959) in that the CCN is more falsifiable, since it predicts a narrower range of outcomes, and therefore has more content. A second reason comes from a Bayesian perspective. Since behaviour in the overlap region is given a greater probability by the CCN rather than the SRN, finding evidence within this region would support the CCN. Conversely, of course, finding behaviour within the SRN populated region would competitively support the SRN over the CCN.

4.2.3.2. Experiment 1: relative frequency prediction conflict. The computer simulations are separated well by this version of the prediction conflict test. Competitive evidence for one model or the other could therefore be obtained by testing people, to see if they are susceptible to this prediction conflict or not.

Method

Subjects. Eighteen undergraduates or recent graduates from the University of Sussex were each paid £1.50 for their co-operation.

Materials and procedure. The experiment had two conditions, with nine subjects completing the conflict condition and nine subjects completing the control. The experiment was controlled and presented by computer with subjects working in isolation.

Subjects were informed of a pending memory test and then presented the training set. Each exemplar from the set was presented one at a time for a duration equivalent to three quarters of a second per character with the whole set repeated five times in total. Subjects were then informed of the existence of a set of rules determining letter order and asked to classify a new set of exemplars as following the rules or not. The test set was presented once, one exemplar at a time with each exemplar shown for a duration equivalent to three quarters of a second per character. After each test exemplar subjects were asked to classify it as either grammatical or ungrammatical and then rate their confidence in that decision on a scale from zero to five where zero meant complete guess and five meant absolutely sure.

Results

The average scores for both conditions, with 95% confidence intervals, are presented in Table 5. The control condition produced significant learning (compared to a baseline of 50%), $t(8) = 5.39$, $p = .00066$, showing that people had acquired a knowledge of the bigram structure in their training language. The conflict condition, however, did not produce performance significantly different from 50%, $t(8) = 0.67$, $p = .53$, suggesting that people had no real favour for exemplars containing either conflict or non-conflict bigrams. The conditions were significantly different, $t(16) = 2.85$, $p = .012$.

This result implies that people's classification is not sensitive to bigram prediction conflicts contained within the classified exemplars. Plotting the confidence interval on the simulation scatter plot (Fig. 6) shows that these results sit firmly in the area occupied by CCN performance.

Table 5
Average performance with confidence intervals on the control and conflict conditions of the relative frequency prediction conflict experiment

	Control	Conflict
+95% CI	60.77	53.44
Average (%)	58.93	51.39
SD	4.97	6.26
−95% CI	57.09	49.34

Discussion

As the scatter plot shows, people's performance is better simulated by the CCN than the SRN we implemented. Not one of the SRN's we ran could produce a behaviour inside the confidence limits of people's performance. On this basis, it appears that people really are not sensitive to prediction conflict frequency in grammar classification tasks. The current SRN simulation did not predict people's behaviour.

4.2.4. Retroactive interference

The second experiment based on prediction conflict bigrams exploits the notion of "catastrophic forgetting." The development, simulation trials and subsequent experiment based on this suggestion is described below.

To test retroactive interference, a training language must present uneven distributions of both prediction conflict and non-prediction conflict bigrams. The strongest implementation of this would involve the first half of training containing one set of experimental bigrams and

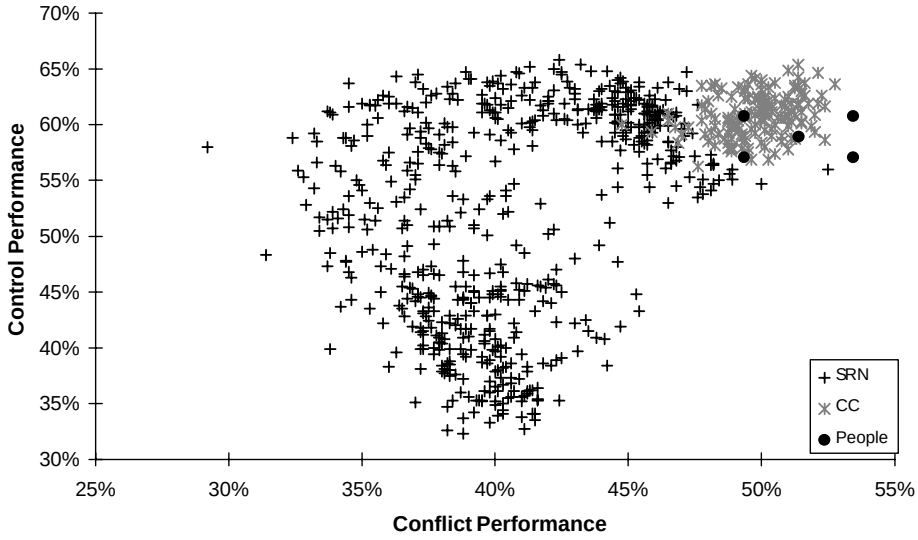

Fig. 6. SRN, CCN and people's performance on the relative frequency prediction conflict experiment. The plot contrasts performance on the control and conflict conditions.

Table 6
Grammars (in the form of rewrite rules) for two training conditions in the retroactive interference experiment

Primary	Conflict secondary	Non-conflict secondary
	Primary then	Primary then
([] → [1])	([] → [7])	([] → [7])
([] → [2])	([] → [8])	([] → [8])
([1] → T[3])	([7] → T[9])	([7] → T[9])
([2] → T[1])	([8] → T[7])	([8] → T[7])
([2] → N[3])	([8] → N[9])	([8] → N[9])
([3] → X[2])	([9] → V[8])	([9] → V[8])
([3] → BV[4])	([9] → BX[10])	([9] → PX[10])
([4] → T[5])	([10] → T[11])	([10] → T[11])
([4] → N[6])	([10] → N[12])	([10] → N[12])
([5] → T[6])	([11] → T[12])	([11] → T[12])
([6] → V[4])	([12] → X[10])	([12] → X[10])
([4] →)	([10] →)	([10] →)
([5] →)	([11] →)	([11] →)
([6] →)	([12] →)	([12] →)

Numbers in brackets refer to states and letters are terminal elements.

the second half containing a second set. The stimuli below achieve this in two conditions: "BV" followed by "BX"; and "BV" followed by "PX." The first of these conditions creates the opportunity for prediction conflict bigram forgetting (retroactive interference), and the second creates a similar opportunity for non-prediction conflict bigrams. The forgetting for prediction conflict bigrams will be compared with the forgetting for non-prediction conflict bigrams.

Table 6 shows the training grammars. Note that for simplicity of exposition in this case, the grammars are shown as rewrite rules. The rewrite rules for each grammar define a finite-state grammar, and so could be represented in the same diagrammatic way as Fig. 4. Notice that the only difference between the secondary grammars is in the second rule rewriting state [9] for each grammar: state [9] goes to BX (and state [10]) in the conflict grammar, but goes to "PX" (and state [10]) in the non-conflict grammar.

Each of the two training languages contained thirty exemplars between three and six letters long. The first fifteen exemplars in each language are the same, generated as a random sample from the primary grammar. The second fifteen are the same random sample, but this time from the secondary conflict and secondary non-conflict grammars. These are shown in Table 7. The conflict condition is called "BVBX" and the non-conflict condition is called "BVPX."

In Experiment 2, a direct test of bigram knowledge was implemented. Consider the bigram "BV" which only appears in the first half of both conditions' training languages. In one condition, the second half of the test set contains a conflicting bigram, "BX," which will affect the knowledge of "BV" in any system sensitive to prediction conflict forgetting. In any such system, there should be a significant decrease in the considered grammaticality of "BV" between the two conditions, and for any systeminsensitive to prediction conflict forgetting, there should be no difference between conditions. A simple test of that bigram's grammaticality is needed. This test should be embedded in a number of other similar tests so as to familiarise a

Table 7
Training languages for the retroactive interference experiment

Primary	Followed by Secondary	
"BV"	Conflict "BX"	Non-conflict "PX"
TBV	TBX	TPX
NBVNV	NBXNX	NPXNX
NBVNVN	NBXNXN	NPXNXN
NBVTT	NBXTT	NPXTT
TBVNVT	TBXNXT	TPXNXT
NBV	NBX	NPX
NBVTTV	NBXTTX	NPXTTX
TTXNBV	TTVNBX	TTVNPX
NBVN	NBXN	NPXN
NXNBV	NVNBX	NVNPX
TTBV	TTBX	TTPX
TBVNV	TBXNX	TPXNX
TXTTBV	TVTTBX	TVTTPX
TTBVTT	TTBXTT	TTPXTT
NBVT	NBXT	NPXT

subject to this task. To accomplish this, in Experiment 2, "grammaticality' ratings were taken for each of the "B_" bigrams: "BT," "BV," "BX," "BB," "BN" and "BP."

4.2.4.1. Computer simulations. To satisfy the new test constraints, the simulations were extended to enable bigram "grammaticality" rating. The simplest option was taken: namely, to let both the SRN and CCN treat these bigrams as new exemplars in need of classification—including hidden start and end markers for the SRN (as was used in the previous simulation and this simulation for all test and training exemplars)—and record the level of probability of classification. Evidence from Shanks (1990) in which people treated direct questions about a classification cue, as a classification task in which only that cue occurred, could be used to support using a probability of classification output from the models.

The two simulations were tested on the two conditions over their specified ranges of parameter settings for one and two training set presentations (epochs). The average probability-of-classification output for each test bigram was recorded for each of these parameter sets, and the results are summarised below.

The average probabilities were normalised for each SRN and CCN parameter set using formula (6) to produce a relative probability, p'. This transformed the models' outputs into a comparable range between zero and one, where zero represented the lowest probability a system gave any bigram and one represented the highest.[4]

$$p'_{(i)} = \frac{p_{(i)} - p_{\min}}{p_{\max} - p_{\min}} \tag{6}$$

$p'_{(i)}$ is the relative probability for bigram i; $p_{\max}$ is the maximum probability assigned any bigram; $p_{\min}$ is the minimum probability assigned any bigram.

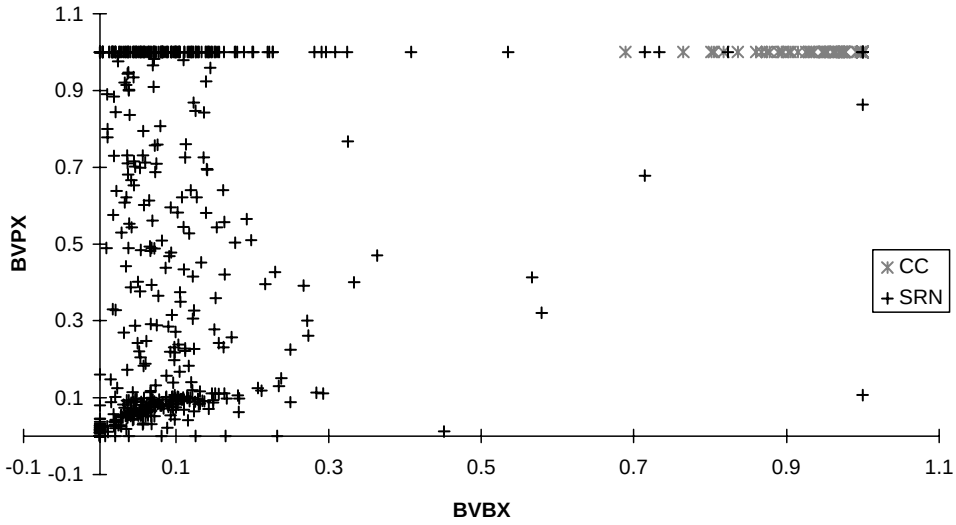

Fig. 7. A scatter plot of P′ (BV), the relative probability assigned to 'BV', under the BVBX condition against the BVPX condition for every SRN and CCN parameter set.

The primary interest with this experiment is in the relative probabilities ascribed the "BV" bigram under the two conditions, BVBX and BVPX. The scatter plot (Fig. 7) shows the relative probabilities for "BV" under each condition for every SRN and CCN parameter set. As can be seen, the two systems form reasonably distinct clusters.

Consider the CCN performance first. Under the BVPX condition, we see that the CCN assigned "BV" a maximum probability for every parameter set. This shows us that it learns bigrams consistently and that, without any conflicting factors, the CCN is not affected by forgetting over a short time period. Under the BVBX condition, the CCN assigned "BV" a high relative probability—on average .97—but not always the maximum probability with every parameter set. "BV" is forgotten slightly in both conditions—as its chunk strength decays—but is only detected as being forgotten when the CCN has a more recent bigram to compare with it, "BX." So, "BV" is not always given the highest probability because of the more recent exposure to "BX." This is the degree of baseline forgetting (i.e., forgetting in the absence of retroactive interference) against which we must compare the SRN.

The SRN's performance is much more varied than that of the CCN, but falls into two clusters. These are separated by performance on the BVPX condition. In this condition, a number of parameter settings produced good bigram learning, allowing the SRN to assign "BV" maximum probability, and a number produced very poor bigram learning with the SRN favouring "B_" bigrams over "BV"—probably reflecting the high frequency of occurrence of those other letters. These are the two performance clusters. Under the BVBX condition, the SRN relative probabilities are nearly all close to zero—on average .082—even when performance on the BVPX condition is high. As predicted, this is the characteristic catastrophic forgetting result for the SRN.

In summary, there is a marked difference between the two models. The distinction is not completely parameter free, since some of the SRN parameter settings did produce CCN like behaviour.

4.2.4.2. Experiment 2: retroactive interference. Method

Subjects. Thirty-six undergraduates and recent graduates from the University of Sussex, were each paid £1.50 for their co-operation.

Materials and procedure. Twelve subjects completed each of the two conditions: "BVBX" and "BVPX." The experiment was run by computer. Subjects were first informed of an impending memory test and then presented the exemplars from their respective training languages one at a time for a duration of three quarters of a second per character. The training language was presented once.

Subjects were then informed of the existence of a complex set of rules determining the order of letters in each exemplar and asked to report how likely they thought it was that the forthcoming pairs of letters would be allowed by those rules. The computer then presented the letter pairs from the training set one at a time, and in a newly randomised order for each subject. After each pair, subjects were asked to respond with an integer from "1" to "10," where "1" meant there was no chance of that bigram being allowed by the rules, "10" meant that the bigram was definitely allowed, "5" meant that the bigram was probably not allowed and "6" meant that the bigram was probably allowed.

The 12 control subjects were just given the test phase. They were told an experiment would be run in which people would have to memorise strings of letters generated by a complex set of rules. They were told that the experimenter was interested in whether people had pre-existing expectations regarding what ordering of letters is likely to be used in such an experiment.

Results

The results are summarised in Table 8 and Fig. 8.

Both the BVBX and the BVPX conditions learnt about the presence of the BV bigram: the BVBX group rated the BV bigram as more likely to be allowed by the rules than the control subjects did, $t(22) = 2.84$, $p = .0096$; similarly, The BVPX group rated the BV bigram as more

Table 8
People's average relative responses (scaled by dividing by 10) to the test bigrams in the retroactive interference experiment

	BVBX	BVPX	Control
BT	0.490 (0.412)	0.432 (0.366)	0.583 (0.204)
BX	0.785 (0.309)	0.594 (0.350)	0.483 (0.269)
BV	**0.823 (0.321)**	**0.830 (0.360)**	**0.483 (0.262)**
BB	0.114 (0.168)	0.097 (0.194)	0.558 (0.250)
BN	0.759 (0.196)	0.616 (0.279)	0.625 (0.114)
BP	0.028 (0.066)	0.266 (0.307)	0.558 (0.204)

Standard deviations appear in parentheses. The experimental bigram line, "BV" is bold.

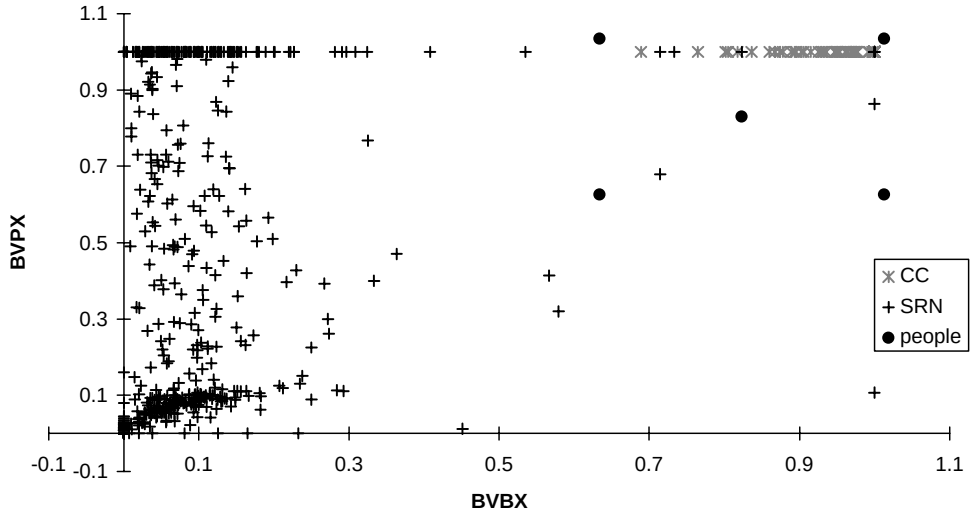

Fig. 8. A scatter plot of P′ (BV), the relative probability assigned to 'BV', under the BVBX condition against the BVPX condition for the SRN, CCN and people.

likely to be allowed by the rules than the control subjects did, $t(22) = 2.70$, $p = .0065$. This bigram received the highest grammatical rating in both experimental conditions. Crucially, there was no significant difference in the "BV" responses between the BVBX and BVPX conditions, $t(22) = 0.05$, $p = .96$. These results show no effect of catastrophic forgetting.

4.2.4.3. Summary. In this experiment, there was evidence of catastrophic forgetting in the SRN. This effect is produced by an uneven distribution of prediction conflict bigrams but is not consistent throughout all parameter settings. In contrast, the CCN is consistently unaffected by this factor. Experiments on people found no catastrophic forgetting and produced results that fell squarely in the performance region occupied by the CCN. On either a Popperian or a Bayesian perspective, the CCN is strongly competitively supported over the SRN.

5. Summary and discussion

In this paper, a number of aspects of bigram sensitivity in the artificial grammar-learning domain has been considered. Stimuli manipulations in two experiments have successfully exposed both characteristic and quantitative differences between the CCN and SRN simulations. For these stimuli, people proved to be both characteristically and quantitatively more like the CCN we implemented than the SRN we implemented. The first experiment manipulated conditional probability, or prediction conflicts. For example, in the training phase, "D" was followed twice as frequently by "N" rather than "V." The SRN was sensitive to this manipulation, with it preferring stimuli containing "DN" rather than "DV" in the test phase. On the other hand, the CCN and people both remained unaffected by the manipulation, even though both had acquired knowledge of the permissible bigrams in the training stimuli. In the second

experiment, the effect of forgetting was investigated, contrasting retroactive interference with simple forgetting; that is, exposure to "DN" followed by exposure to "DV" produces retroactive interference of the association between "D" and the subsequent "N"; exposure to "DN" followed by exposure to "BV" results in simple bigram forgetting of "DN." It was predicted that the SRN compared to the CCN should show greater retroactive interference in comparison to simple forgetting. This is just what was found. People's behaviour was well modelled by the CCN and poorly modelled by the SRN, a result in accordance with that of the first experiment. Thus, both experiments competitively support the CCN over the SRN as a model of learning associations in artificial grammar learning.

The CCN and SRN differ on two main dimensions. First, the CCN is an example of a localist network and the SRN is an example of a network that learns by acquiring largely distributed representations of higher order structure. Localist coding provides some robustness to retroactive interference precisely because changes in representation are localized to particular representations. With distributed representations, learning about something new changes the representation medium globally, and thereby influences the coding of everything else that has been learnt. Our results favour the notion that people learn chunks by forming (property-structure) explicit chunk representations because learning new chunks interferes little with the knowledge of old chunks (people were not sensitive to prediction conflicts). Second, the CCN learns by simply adding to its knowledge store; the SRN learns by error correction. This is a second reason underlying the stability of the CCN's knowledge. Adding does not involve unlearning; error correction involves unlearning what you have learnt if it now leads to error. Kruschke (1993) showed how error correction with non-local representation leads to catastrophic interference.

We now discuss differences in scope between the SRN and CCN models, the generality of the experimental findings with people, and finally, general principles for evaluating models.

5.1. Scope differences between the CCN and the SRN

Although the CCN outperforms the SRN on these data, the SRN has been used in previous research to deal with a range of different learning situations. We will consider a number of examples, and argue in each case either that the CCN could in principle be extended to cover a similar scope, or that it has a fundamental weakness. From a Popperian perspective, other things being equal, a model with greater scope has greater Popperian content (more domains in which it can be falsified), and hence should be preferred.

The SRN has been used for modelling people's ability on the artificial grammar learning task to learn strings made up of some set of elements (e.g., letters) but classify strings made up of completely different elements (e.g., visual icons) (Dienes et al., 1999). The CCN as it stands cannot do this. Redington and Chater (1996) showed that letter chunks contained sufficient information to enable transfer. Indeed, the CCN can be extended to produce an "Abstract Competitive Chunker" to enable transfer with the mechanisms of competitive chunking, in a similar way as Dienes et al. (1999) extended the SRN to deal with transfer. Letters could be linked to a layer of elementary abstract chunks X1, X2, etc. (just as the Dienes et al. SRN works on a recoding of the front end input). The equations of competitive chunking would then apply to the abstract chunks. On any one stimulus presentation, a given letter becomes bound to a particular elementary abstract chunk (e.g., X1 might be instantiated with M), and

on that presentation X1 must always be instantiated with M (but on presentation of the next stimulus, these bindings are wiped). Each time a letter instantiates a particular elementary abstract chunk, the more higher order chunks that elementary chunk itself takes part in on subsequent passes of the same stimulus presentation, the more likely the letter is to elicit that elementary abstract chunk in future. The more times an abstract elementary chunk has been used, the more likely it is to be elicited at all, by any letter. These two properties would result in the network preferring particular letters in particular chunks, but if forced with a new domain, it would prefer to impose frequently used chunk structures on the domain. These ideas need to be developed further in formal simulations, but they at least show that transfer is not an in principle problem for the mechanisms of competitive chunking.

The SRN has also been used by Kinder and Shanks (2001) to simulate the behaviour of amnesiacs learning artificial grammars. Amnesic people classify old versus new strings relatively poorly compared to normal people, but they classify new grammatical versus new ungrammatical strings at about the same level as normal people (Knowlton & Squire, 1994, 1996). Kinder and Shanks showed that this pattern could be simulated simply by assuming amnesic people have a lower learning rate than normal people. It has yet to be shown whether the CCN could simulate the Knowlton and Squire data. In principle, a faster rate of decay or smaller competition parameter for amesiacs compared to normals are possible. These parameter differences between the populations have a stronger face-validity than a smaller learning rate: a small learning rate means integrating *further* into the past. A priori one might expect normals to integrate over trials further back into the past than amnesiacs, contrary to the learning rate hypothesis. This is a prediction we plan to test directly in future work.

The SRN has been very successful in modelling how people learn sequentially presented information. For example, the SRN has been found to be useful in simulating people's performance on sequential reaction time tasks (Cleeremans & McClelland, 1991). More broadly, the SRN has also been used to capture aspects of language learning, such as the potential importance of starting with a small and increasing working memory in learning progressively more complex grammatical structures (Elman, 1993) and people's (limited) sensitivity to recursion in natural language (Christiansen & Chater, 1999a). Plaut (1999) found the sequential nature of the SRN useful for modelling learning to read individual words, reading being an inherently sequential process of converting graphemes into a sequence of phonemes. Similarly, Gaskell and Marslen-Wilson (1997) found the sequential nature of the SRN useful for modelling speech perception; speech being an inherently sequential process in which a diminishing set of lexical candidates can be activated as a word is heard. While some of the shortcomings of the CCN could be solved by, e.g., entering input to it sequentially, any learning mechanism that simply chunks surface form will be inadequate as a model of language learning (e.g., Chomsky, 1957).[5] Language involves at least a context-free grammar (e.g., Gazdar, Klein, Pullam, & Sag, 1985), approximated to some degree. Relatedly, language behaviour also exhibits systematicity (Fodor & Pylyshyn, 1988) and structure dependency (Chomsky, 1980). The SRN can (imperfectly) learn some context-free grammars (e.g., Christiansen & Chater, 1999a); arguably, it can also develop a qualified systematicity (Christiansen & Chater, 1994; also see Hadley, 1994a,b). Mere chunking of surface forms is exactly the wrong approach for ending up with a device that can acquire (to some degree of approximation) dependence on phrase structure in a systematic way. Research using the SRN has, at least, taken some small

but important steps towards showing it has some properties required for a language-learning device. The challenge for the future is to determine a model more robust to interference than an SRN—like the CCN, as shown in this paper—but also capable of inducing context-free (and higher order) grammars.

5.2. How general are the findings with people?

In contrast to the experimental results in this paper, people in many situations are sensitive to prediction conflicts or relative contingency (Shanks, 1995). For example, if people are asked to predict diseases given symptoms, they will be sensitive to the relative probability of different diseases given an imperfect predictor symptom, appropriately choosing the disease more often that has the highest conditional probability, at least under certain circumstances, a conceptual analogue of our prediction conflict (Medin & Edelson, 1988). On other tasks, people can be sensitive to levels of contingency smaller than those used in the current experiments (e.g., Chatlosh, Neunaber, & Wasserman, 1985). Why should there be a difference from our results? Perhaps people's knowledge of relative contingency in the artificial grammar learning paradigm simply plays little part in classification decisions, just as it plays little part in the CCN. Alternatively, although there may be an underlying mechanism—involving error correction—sensitive to contingency, subjects in addition may refer to a one-shot learning mechanisms for chunks. The SRN just models the error correction component, people's final decision is affected by both mechanisms. Perhaps different amounts of exposure to stimuli lead to different relative contributions from the two mechanisms. If one introduced the additional assumption that the one-shot mechanism was explicit[6] and error-correction an implicit mechanism, such a theory could be tested by using a manipulation which differentially disrupts explicit memory: For example, if subjects performed a secondary task at training or test, it could be determined whether the data still matches the CCN's behaviour better than the SRN's.

Another reason why people may not have appeared sensitive to relative contingency is that in artificial grammar learning people may not be explicitly trying to predict each letter from the last. Maybe different principles apply to incidental learning than to prediction tasks? The most successful model of people's categorization behaviour in tasks that involve the explicit intention to learn to classify (like the disease classification task) involves (1) the storage of particular items (e.g., symptom combinations); and (2) the learning of weights from these items to categories (e.g., diseases) according to prediction error (i.e., like back propagation) (e.g., Estes, 1988; Kruschke, 1992). If subjects are not trying to predict anything, and don't feel a sense of success or failure in prediction, maybe the main form of learning is the storage of particular items. If the items are complex, storage can only occur incrementally, as specified by the CCN. Perhaps if people's explicit task in training were to predict the next letter they would behave more like the SRN, and the probabilities of people generating different letters would be sensitive to the conditional probabilities of those letters.

Relatedly, Seger (1997, 1998) argued that perceptual motor implicit knowledge, as typically assessed in the sequential reaction time task, is learnt by different mechanisms than the implicit knowledge underlying judgement tasks like typical artificial grammar learning. Gomez (1997) speculated that explicit chunking mechanisms may operate best when all the letters of a string are presented simultaneously. Gomez used a sequential artificial grammar learning task in

which only one letter was shown at a time, and found learning could still be obtained as evidenced by faster reaction times to type letters in a grammatical rather than ungrammatical position.

But does presenting one letter at a time compared to all letters at once change subjects' learning? Using the grammar and test stimuli of Dienes et al. (1991), we compared a group of 10 subjects who saw each string in its entirety (as in the previous experiments) for a total duration of 200 ms per letter with a group who saw only one letter at a time, for 200 ms per letter. In training, the subjects tried to memorise the strings silently. In test, stimuli were presented under the same conditions as training for each group and subjects classified strings as being grammatical or not. The "all letters at once" group ($M = 65\%$, $SD = 7.0\%$) classified better than the "one letter at a time" group ($M = 56\%$, $SD = 4.3\%$), $t(18) = 3.00$, $p = .0076$. This suggests that under standard conditions of artificial grammar learning, people use a learning device—potentially like the CCN—that benefits by the whole string being displayed at once. Future research could investigate what happens if people try to predict each letter as it comes up sequentially, and whether sensitivity to prediction conflicts then emerges; i.e., if people come to behave more like the SRN. If not, the scope of the CCN will have been broadened. Further, auditory stimuli are by their nature sequential. The SRN might be a better model of learning finite state grammars with auditory stimuli (as, e.g., used by Altmann, Dienes, & Goode, 1995). Dienes and Longuet-Higgins (submitted) investigated implicit learning of musical sequences involving non-local contingencies (e.g., relationships like transposition or inversion); the CCN, therefore would not be able to model these data.

5.3. *Evaluating models*

On a more general level, this paper has illustrated how connectionist models can be part of a genuine scientific exploration of human psychology. One criticism of connectionist models is that if the network is just a black box, it is unclear how much has been achieved in the way of explanation—just what about the simulation was important in creating a good fit (McCloskey, 1991)? Our approach has been to define plausible networks, based on different core assumptions regarding learning and representation: In the case of the SRN, the basic assumption is that learning a sequence proceeds by attempting to predict the next element, and building up a distributed representation of proceeding context as it is needed; in the CCN, the basic assumption is that (property-structure) explicit fragments of the sequence are stored and gradually chunked together. Implementing these assumptions requires the use of various free parameters, and this is where criticisms of connectionism being too powerful need addressing. According to this type of criticism, connectionist networks have the power to approximate any computational function, so finding that some network can simulate a set of data is not useful—it is a foregone conclusion (Massaro, 1988). (This criticism is just as powerfully applied to symbolic models—it is also a foregone conclusion that there is some symbolic model that can simulate a set of data.) While this criticism has already been amply rebutted by the wealth of specific connectionist models that have since been produced that have *differentially* good fits to different data sets (see, e.g., Christiansen & Chater, 1999b; McLeod, Plunkett, & Rolls, 1998), we used an approach that while partly implicit in the practice of many researchers, it is not made explicit in most published reports on the application of connectionist networks

to psychology. We considered the space of dependent variables by which the model's performance is compared to human data. The regions of this space occupied by the models when their parameters are allowed to take a plausibly full range of values are compared. These regions help define the characteristic behaviour of the models, regardless of the existence of some non-characteristic performances with some parameter values. At a qualitative level, human behaviour can be assessed as to whether it is more like the characteristic behaviour of one model or the other. More quantitatively, if, as we actually found, the models have overlapping regions, but one model occupies a smaller region in total, then both Bayesian and Popperian considerations would lead human data in that region to favour the model with the smallest region (which may have the fewest parameters, but it may not).

In Bayesian terms, this could be made more precise, by considering the probability density $p_i = p(\text{data}/\text{model } i)$ (see Mackay, 1995, for application of Bayesian ideas to model selection). $p(\text{data}/\text{model } i)$ is the integral over parameters of the probability of the data for a particular parametric specification of the model times the prior probability of the parameters. Then p_i/p_j would give the factor by which the ratio of the prior probabilities of the models being correct should be multiplied to get the ratio of the posterior probabilities. In Bayesian terms, the particular approach adopted in this paper of considering the behaviour of each model over a specified range of parameters amounts to assuming a uniform prior for parameter values over the specified range, with negligible probability of the parameter outside of the range. This is a useful simplifying assumption, but it needs to be considered carefully in each application. While sometimes there may be strong *a priori* reasons for restricting the range to some value (e.g., the *d* parameter for the CCN), sometimes a more or less arbitrary decision was made (e.g., the *c* parameter for the CCN). The conclusions drawn are of course conditional on the range used; future considerations might lead to a different parameter range and a revision of conclusions. The properties of the models explored in this paper were also derived on theoretical grounds; they motivated the simulations and were not just derived *post hoc* from the simulations. Therefore, we can be relatively confident that the characteristic behaviour of the models is as the simulations suggested, even given some arbitrariness in the choices of parameter ranges.

One need not buy into the Bayesian approach in toto to appreciate the general logic that the evidence favours the model that most strongly predicts it. As long as one appreciates this logic, there is a means of choosing between competing models, even though both have various free parameters, and even if both COULD simulate the human data with some parameter values to within the limits of error of the experimental data. A common alternative way of choosing between models, or simply promoting a single model, is to determine the parameters that produce the best fit to the data (e.g., Dienes et al., 1999). This procedure for fitting models automatically favours complex models (e.g., those with more parameters, or with more flexible functional forms, Myung & Pitt, 1997); choosing models according to which most strongly predicts the data automatically favours simple models, those with fewer parameters, or which are less parameter sensitive (Mackay, 1995; Popper, 1959). Myung and Pitt showed that quantitatively considering the probability of the obtained data given each model leads to more reliable identification of the true underlying model than simply using the model with the best fit to the data (even for models with the same number of parameters). In summary, an explicit and systematic consideration of the behaviour of a model over its parameter space is important to fully assess the worth of a model.

We have attempted to provide analyses of why the models characteristically behave as they do, thus they are not simply black boxes. The problems of pinpointing the cause of the models' success (if such a question is even strictly meaningful) is not unique to computational modelling. Theoretical explanation in all domains starts from the core assumptions of the theory. But to apply it in any particular case, auxiliary satellite assumptions are needed. The whole network of assumptions bears some of the inferential weight in making predictions. If, in successive applications to different domains, the core assumptions can be retained, they prove their scientific value. We don't see computational modelling as any different in this respect.

6. Conclusion

In conclusion, in this paper we explored one of the most basic, ubiquitous and important of people's learning abilities: the ability to learn a relationship between two items, to form a chunk or an association. The artificial grammar-learning paradigm was used to explore this ability because learning in this paradigm predominantly involves learning chunks. The relative resistance of people's chunk knowledge to interference supports a role for models that use localist incremental learning like the CCN. The CCN cannot be a complete account of all implicit learning, e.g., it is not complete in language and perhaps not in music (Dienes & Longuet-Higgins, submitted), but the CCN is surely modelling a very important learning ability nonetheless.

For further reading

The following references may also be of interest to the readers: Boucher (1998), Chan (1992), Chandrasekaran, Goel, and Allemang (1988), Dienes and Perner (forthcoming), Grossberg (1987), Marr (1982), Melz, Cheng, Holyoak, and Waldmann (1993), Seger (1994), Shanks (1993), Shanks and St. John (1994), and Waldmann and Holyoak (1992).

Notes

1. In fact, whether back propagation was left on in the test phase or not made no difference to the pattern of results.
2. This is not a completely faithful interpretation of competitive chunking. The differences are listed at the end of this section.
3. The restriction on overlapping chunks could be implemented using lateral inhibitory connections between units representing overlapping chunks. However, this makes the process of growing new chunks more complicated. Thus, while the current method is admittedly *ad hoc* and simplistic, and a realistic model would use lateral inhibition, the current method's simplicity gives greater clarity in understanding the model's behaviour.
4. This measure separated out the behaviour of the two models neatly; it is simply a way of rescaling model probabilities, so it produces output just as "raw" as any other way of producing probabilities from a model.

5. Perruchet and Vinter (in press) briefly suggest learning by chunking could apply to all incidental learning, even to natural language acquisition, if the content of chunks can contain abstract structures. They further suggest that the content of the chunks are conscious. Putting aside the cavalier nature of their response to the problems of linguistics and computational linguistics, the two claims of Perruchet and Vinter put them on the horns of an impossible dilemma. On the one hand, to capture the complexity and productivity of language, the chunks must contain very abstract structures. (e.g., suppose we take a "chunk" to be a lexical entry. Lexicalist models of syntax, e.g., Bates & Goodman, 1997, involve putting more and more of the syntax into the lexicon and thus the lexicon contains many complex phrase structures and productive rules.) On the other hand, to capture our deep lack of conscious awareness concerning how we understand and produce language, the content of the chunks cannot be very abstract at all. (We are clearly not aware of the phrase structures and productive rules postulated by lexicalist accounts to exist in the lexicon.).
6. Contrast Dienes and Fahey (1998) where chunks (in a different implicit learning paradigm) were argued to be based on implicit memory. However, knowledge of chunks is often argued to be explicit, e.g., Dulany et al. (1984), Perruchet and Pacteau (1990).

References

Altmann, G. T. M. (in press). Learning and development in neural networks—the importance of prior experience. *Cognition*.

Altmann, G. T. M., & Dienes, Z. (1999). Rule learning by seven-month-old infants and neural networks. *Science*, *284*, 875.

Altmann, G., Dienes, Z., & Goode, A. (1995). On the modality independence of implicitly learned grammatical knowledge. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*, 899–912.

Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah: Lawrence Erlbaum.

Aslin, R. N., Saffran, J. R., & Newport, E. L. (1998). Computation of conditional probability statistics by 8-month-old infants. *Psychological Science*, *9*, 321–324.

Bates, E., & Goodman, J. (1997). On the inseparability of grammar and the lexicon: Evidence from acquisition, aphasia, and real-time processing. *Language and Cognitive Processes*, *12*, 507–584.

Berry, D. C. (Ed.) (1997). *How implicit is implicit learning?* Oxford: Oxford University Press.

Berry, D. C., & Dienes, Z. (1993). *Implicit learning: Theoretical and empirical issues*. Hove: Lawrence Erlbaum.

Bishop, C. M. (1996). *Neural networks for pattern recognition*. Oxford: Clarendon Press.

Boucher, L. (1998). *Learning the structure of artificial grammars: Computer simulations and human experiments*. Unpublished D.Dhil thesis, University of Sussex.

Broeder, P., & Plunkett, K. (1994). Connectionism and second language acquisition. In N. Ellis (Ed.), *Implicit and explicit learning of languages* (pp. 421–453). London: Academic Press.

Buchner, A. (1994). Indirect effects of synthetic grammar learning in an identification task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*, 550–566.

Chan, C. (1992). *Implicit cognitive processes: Theoretical issues and applications in computer systems design*. Unpublished D.Phil. thesis, University of Oxford.

Chandrasekaran, B., Goel, A., & Allemang, D. (1988). Information processing abstractions: The message still counts more than the medium. *Behavioural and Brain Sciences*, *11*, 26–27.

Chatlosh, D. L., Neunaber, D. J., & Wasserman, E. A. (1985). Response-outcome contingency: Behavioural and judgmental effects of appetitive and aversive outcomes with college students. *Learning & Motivation*, *16*, 1–34.

Chomsky, N. (1957). *Syntactic structures*. The Hague: Mouton.
Chomsky, N. (1980). *Rules and representations*. Oxford: Basil Blackwell.
Christiansen, M. H., & Chater, N. (1994). Generalization and connectionist language learning. *Mind & Language*, *9*, 273–287.
Christiansen, M. H., & Chater, N. (1999a). Connectionist natural language processing: The state of the art. *Cognitive Science*, *23*, 417–437.
Christiansen, M. H., & Chater, N. (1999b). Toward a connectionist model of recursion in human linguistic performance. *Cognitive Science*, *23*, 157–205.
Cleeremans, A. (1993). *Mechanisms of implicit learning: Connectionist models of sequence processing*. Cambridge, MA: MIT Press.
Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, *120*, 235–253.
Cleeremans, A., Destrebecqz, A., & Boyer, M. (1998). Implicit learning: News from the front. *Trends in Cognitive Sciences*, *2*, 406–415.
Cleeremans, A., Servan-Schreiber, D., & McClelland, J. L. (1989). Graded state machines: The representation of temporal contingencies in feedback networks. In Y. Chauvin & D. E. Rumelhart (Eds.), *Backpropagation: Theory, architectures, and applications.* Hillsdale, NJ: Lawrence Erlbaum.
Dienes, Z. (1992). Connectionist and memory array models of artificial grammar learning. *Cognitive Science*, *16*, 41–79.
Dienes, Z. (1993). Computational models of implicit learning. In D. C. Berry & Z. Dienes (Eds.), *Implicit learning: Theoretical and empirical issues*. Hove: Lawrence Erlbaum.
Dienes, Z., & Fahey, R. (1998). The role of implicit memory in controlling a dynamic system. *Quarterly Journal of Experimental Psychology*, *51A*, 593–614.
Dienes, Z., & Longuet-Higgins, H. C. (submitted). *Can the structures of serialist music be implicitly learnt*?
Dienes, Z., & Perner, J. (1996) Implicit knowledge in people and connectionist networks. In G. Underwood (Ed.), *Implicit cognition* (pp. 227–256). Oxford: Oxford University Press.
Dienes, Z., & Perner, J. (1999). A theory of implicit and explicit knowledge. *Behavioural and Brain Sciences*, *22*, 735–755.
Dienes, Z., & Perner, J. (2002a). A theory of the implicit nature of implicit learning. In A. Cleeremans & R. French (Eds.), *Implicit learning* (pp. 68–92). Hove, England: Psychology Press.
Dienes, Z., & Perner, J. (2002b). The metacognitive implications of the implicit–explicit distinction. In P. Chambres, M. Izaute, & P.-J. Marescaux (Eds.), *Metacognition: Process, function, and use* (pp. 241–268). Dordrecht, The Netherlands: Kluwer Academic Publishers.
Dienes, Z., & Perner, J. (forthcoming). Unifying consciousness with explicit knowledge. In A. Cleeremans (Ed.), *The unity of consciousness: Binding, integration, and dissociation.* Oxford: Oxford University Press.
Dienes, Z., Altmann, G., & Gao, S.-J. (1999). Mapping across domains without feedback: A neural network model of implicit learning. *Cognitive Science*, *23*, 53–82.
Dienes, Z., Altmann, G., Kwan, L., & Goode, A. (1995). Unconscious knowledge of artificial grammars is applied strategically. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *21*, 1322–1338.
Dienes, Z., Broadbent, D. E., & Berry, D. C. (1991). Implicit and explicit knowledge bases in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*, 875–882.
Dienes, Z., Kurz, A., Bernhaupt, R., & Perner, J. (1997). Application of implicit knowledge: Deterministic or probabilistic? *Psychologica Belgica*, *37*, 89–112.
Dulany, D. E., Carlson, R., & Dewey, G. (1984). A case of syntactical learning and judgement: How concrete and how abstract? *Journal of Experimental Psychology: General*, *113*, 541–555.
Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.
Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, *48*, 71–99.
Estes, W. K. (1988). Toward a framework for combining connectionist and symbol processing models. *Journal of Memory and Language*, *27*, 196–212.
Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3–71.

French, R. M., & Cleeremans, A. (2002). *Implicit learning and consciousness: An empirical, philosophical and computational consensus in the making?* Hove, England: Psychology Press.

Gaskell, G. M., & Marslen-Wilson, W. D. (1997). Integrating form and meaning: A distributed model of speech perception. *Language and Cognitive Processes*, *12*, 613–656.

Gaskell, G. M., & Marslen-Wilson, W. D. (1999). Ambiguity, competition, and blending in spoken word recognition. *Cognitive Science*, *23*, 439–462.

Gazdar, G., Klein, E., Pullum, G., & Sag, I. (1985). *Generalized phrase structure grammar*. Oxford: Blackwell.

Gluck, M. A., & Bower, G. H. (1988). From conditioning to category learning: An adaptive network model. *Journal of Experimental Psychology: General*, *117*, 227–247.

Gomez, R. L. (1997). Transfer and complexity in artificial grammar learning. *Cognitive Psychology*, *33*, 154–207.

Gordon, P., & Holyoak, K. J. (1983). Implicit learning and generalisation of the "mere exposure effect". *Journal of Personality and Social Psychology*, *45*, 492–500.

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, *11*, 23–63.

Hadley, R. F. (1994a). Systematicity in connectionist language learning. *Mind & Language*, *9*, 247–272.

Hadley, R. F. (1994b). Systematically revisited. *Mind & Language*, *9*, 431–444.

Johnstone, T., & Shanks, D. R. (1999). Two mechanisms in implicit artificial grammar learning? Comment on Meulmans and Van der Linden (1997). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*, 524–531.

Kinder, A., & Shanks, D. R. (2001). Amnesia and the declarative/procedural distinction: A recurrent network model of classification, recognition, and repetition priming. *Journal of Cognitive Neuroscience*, *13*, 648–669.

Knowlton, B. J., & Squire, L. R. (1994). The information acquired during artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*, 79–91.

Knowlton, B. J., & Squire, L. R. (1996). Artificial grammar learning depends on implicit acquisition of both abstract and exemplar-specific information. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 169–181.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review*, *99*, 22–44.

Kruschke, J. K. (1993). Human category learning: Implications for backpropagation models. *Connection Science*, *5*, 3–36.

Kruschke, J., & Johansen, M. K. (1999). A model of probabilistic category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*, 1083–1119.

Mackay, D. J. C. (1995). Probable networks and plausible predictions—A review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, *6*, 469–505.

Marr, D. (1982) *Vision*. San Francisco: W.H. Freeman and Company.

Massaro, D. (1988). Some criticisms of connectionist models of human performance. *Journal of Memory and Language*, *27*, 213–234.

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science*, *2*, 387–395.

McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In G. Bower (Ed.), *The psychology of learning and motivation* (Vol. 24). New York: Academic Press.

McLeod, P., Plunkett, K., & Rolls, E. T. (1998). *Introduction to connectionist modelling of cognitive processes*. Oxford: Oxford University Press.

Medin, D. L., & Edelson, S. (1988). Problem structure and the use of base-rate information from experience. *Journal of Experimental Psychology: General*, *117*, 68–85.

Melz, E. R., Cheng, P., Holyoak, K. J., & Waldmann, M. R. (1993). Cue competition in human categorisation: Contingency or the Rescorla–Wagner learning rule? Comment on Shanks (1991). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *19*, 1398–1410.

Meulmans, T., & Van der Linden, M. (1997). Associative chunk strength in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *23*, 1007–1028.

Myung, I. J., & Pitt, M. A. (1997). Applying Occam's Razor in modelling cognition: A Bayesian approach. *Psychonomic Bulletin & Review*, *4*, 79–95.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard.

Nosofsky, R. M., Palmeri, T. J., & McKinley, S. C. (1994). Rule-Plus-Exception model of classification learning. *Psychological Review, 101*, 53–79.

Page, M. P. A. (2000). Connectionist modelling in psychology: A localist manifesto. *Behavioural and Brain Sciences, 23*, 443–512.

Perruchet, P., & Pacteau, C. (1990). Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *Journal of Experimental Psychology: General, 119*, 264–275.

Perruchet, P., & Vinter, A. (1998). PARSER: A model for word segmentation. *Journal of Memory and Language, 39*, 246–263.

Perruchet, P., & Vinter, A. (in press). The self-organizing consciousness. *Behavioural and Brain Sciences*.

Plaut, D. C. (1999). A connectionist approach to word reading and acquired dyslexia: Extension to sequential processing. *Cognitive Science, 23*, 543–568.

Popper, K. (1959). *The logic of scientific discovery*. London: Hutchinson.

Reber, A. S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behaviour, 6*, 855–863.

Reber, A. S. (1989). Implicit learning and tacit knowledge. *Journal of Experimental Psychology: General, 118*, 219–235.

Reber, A. S. (1992). The cognitive unconscious: An evolutionary perspective. *Consciousness and Cognition, 1*, 93–133.

Redington, M., & Chater, N. (1996). Transfer in artificial grammar learning: A re-evaluation. *Journal of Experimental Psychology: General, 125*, 123–138.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge: Cambridge University Press.

Seger, C. A. (1994). Implicit learning. *Psychological Bulletin, 115*, 163–196.

Seger, C. A. (1997). Two forms of sequential implicit learning. *Consciousness and Cognition, 6*, 108–131.

Seger, C. A. (1998). Independent motor-linked and judgment-linked forms of artificial grammar learning. *Consciousness and Cognition, 7*, 259–284.

Servan-Schreiber, E., & Anderson, J. R. (1990). Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 16*, 592–608.

Shanks, D. R. (1990). Connectionism and the learning of probabilistic concepts. *The Quarterly Journal of Experimental Psychology, 42A*, 209–237.

Shanks, D. R. (1993). Associative versus contingency accounts of category learning: Reply to Melz, Cheng, Holyoak, and Waldmann (1993). *Journal of Experimental Psychology: Learning, Memory, and Cognition, 19*, 1411–1423.

Shanks, D. R. (1995). *The psychology of associative learning*. Cambridge, MA: Cambridge University Press.

Shanks, D., & St. John, R. (1994). Characteristics of dissociable human learning systems. *Behavioural and Brain Sciences, 17*, 367–447.

Shanks, D. R., Charles, D., Darby, R. J., & Azmi, A. (1998). Configural processes in human associative learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 24*, 1353–1378.

Waldmann, M. R., & Holyoak, K. J. (1992). Predictive and diagnostic learning within causal models: Asymmetries in cue competition. *Journal of Experimental Psychology: General, 121*, 222–236.