COGNITIVE SCIENCE Vol 23 (1) 1999, pp. 53–82 ISSN 0364-0213

Mapping across Domains Without Feedback: A Neural Network Model of Transfer of Implicit Knowledge

ZOLTÁN DIENES
University of Sussex, UK

GERRY T. M. ALTMANN
University of York, UK

SHI-JI GAO
Research Institute of Economic System and Management, China

This paper shows how a neural network can model the way people who have acquired knowledge of an artificial grammar in one perceptual domain (e.g., sequences of tones differing in pitch) can apply the knowledge to a quite different perceptual domain (e.g., sequences of letters). It is shown that a version of the Simple Recurrent Network (SRN) can transfer its knowledge of artificial grammars across domains without feedback. The performance of the model is sensitive to at least some of the same variables that affect subjects' performance—for example, the model is responsive to both the grammaticality of test sequences and their similarity to training sequences, to the cover task used during training, and to whether training is on bigrams or larger sequences.

I. INTRODUCTION

This paper is concerned with the problem of how a neural network could model the way people can acquire knowledge in one perceptual domain and apply it to a quite different one. The process of forming a mapping between two disparate domains is a fundamental aspect of human cognition—it is central in solving problems, forming insights, even understanding jokes.[1] On the other hand, connectionist models have been criticized precisely because their knowledge is highly inflexible and domain dependent (Clark & Karmiloff-Smith, 1993) . This paper will attempt to show how the knowledge embedded in a connec-

Direct all correspondence to: Zoltán Dienes, Laboratory of Experimental Psychology, Sussex University, Brighton, Sussex, BN1 9QG, England <dienes@biols.susx.ac.uk>.

tionist network trained in one domain can be transferred to another domain with entirely different front-end content.

One area in which people have been shown to have a remarkable ability to transfer knowledge across perceptual domains is in the learning of artificial grammars. We will consider this area as a key paradigm for testing how knowledge embedded in a connectionist network can be flexibly transferred across domains. In the artificial grammar learning paradigm, participants are typically asked to memorize strings of letters generated by a finite state grammar (see Figure 1). This exposure to grammatical strings enables participants, to classify new strings as obeying the rules or not (Reber, 1967). Reber found that after training participants had difficulty verbalizing the rules of the grammar, and concluded that participants had induced knowledge that was both unconscious and abstract. Both these claims generated controversy (see Berry & Dienes, 1993; Seger, 1994; Shanks & St John, 1994 for reviews). The important characteristic of artificial grammar learning for this paper is a further demonstration by Reber (1969) that the knowledge was not strongly tied to specific perceptual features. He asked participants to memorise strings of letters generated by a finite state grammar. The more strings participants had previously studied, the easier they found it to memorize new strings generated by the grammar. This benefit remained intact even when the new strings were constructed from a different letter set, but the same grammar. That is, participants could apply their knowledge of the grammar regardless of the letter set. Similarly, Mathews, Buss, Stanley, Blanchard-Fields, Cho, and Druhan (1989), Brooks and Vokey (1991), Whittlesea and Dorken (1993), Gomez and Schvaneveldt (1994), Manza and Reber (1997), and Shanks, Johnstone, and Staggs (1997) showed that when participants., were exposed to strings constructed from one letter set,

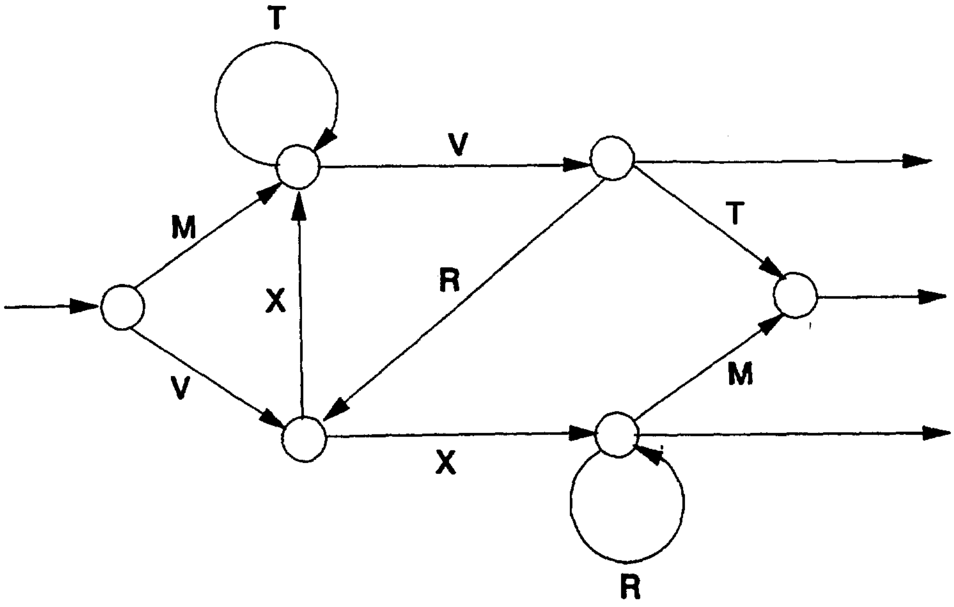

Figure 1. A finite state grammar

they could later classify as grammatical or not strings constructed from another letter set, even though participants, were not told the mapping between the letter sets and were not given feedback on the accuracy of their responses. Altmann, Dienes, and Goode (1995) and Manza and Reber (1997) extended these findings by showing that participants, trained in one modality (e.g., sequences of tones) could classify in another modality (e.g., sequences of letters or arbitrary symbols), even when the mapping between the domains was arbitrary and no accuracy feedback was given.

This paper is concerned with modelling this ability of people to transfer knowledge of an artificial grammar across different domains. We define two domains as being different if the mapping between the domains is not a priori known to participants (or to the model). Our aim is to provide an account of how that mapping could be established in the context of a plausible model of artificial grammar learning. Next we will give a brief overview of existing computational models of artificial grammar learning (see Berry & Dienes, 1993, for a fuller review) before turning to the problem of how transfer could be modelled.

II. COMPUTATIONAL MODELS OF ARTIFICIAL GRAMMAR LEARNING

There have been various computational models of how participants learn and apply knowledge of an artificial grammar in the same domain: Competitive chunking (Servan-Schreiber & Anderson, 1990), exemplar models (Dienes, 1992), classifier systems (Druhan & Mathews, 1989), and connectionist models (Cleeremans & McClelland, 1991; Dienes, 1992).

In most of these models, the knowledge acquired is intimately bound up with the perceptual features of the domain in which learning took place. For example, in the competitive chunking model of Servan-Schreiber and Anderson (1990) learning consists of automatically forming progressively higher-order chunks. The string TTXVPXVS may first be chunked as (TTX), (VP), and (XVS). At the next learning trial, the chunk ([TTX][VP]) may be formed, and finally the whole string could form a single chunk, resulting in it seeming maximally familiar when later perceived. Similarly, training strings in the exemplar models were represented as specific letters in particular positions; knowledge in the connectionist networks (Dienes, 1992) refers to the relations between specific letters at particular positions. Finally, the classifiers (i.e., condition-action rules) used in the classifier system of Druhan and Mathews (1989) also referred to specific letters at particular positions Mathews and Roussel (1997) showed how classifier systems (and also exemplar models) could produce transfer by coding more abstract features. Crucially, the features encoded could be more than specific letter sequences. They could also be an abstract pattern of runs of identical letters (e.g., MTTTV, could be encoded as "—rr-", where "r" stands for a repeat of the immediately preceding letter). That is, the classifier system allowed transfer only in so far as the different domains had similar patterns of runs in each string. We will see below that, in a new domain, participants can correctly classify strings with no repeated letters (adjacent or otherwise, Altmann et al., 1994), indicating that this cannot be the whole story. We will show that a simple extension of the approach

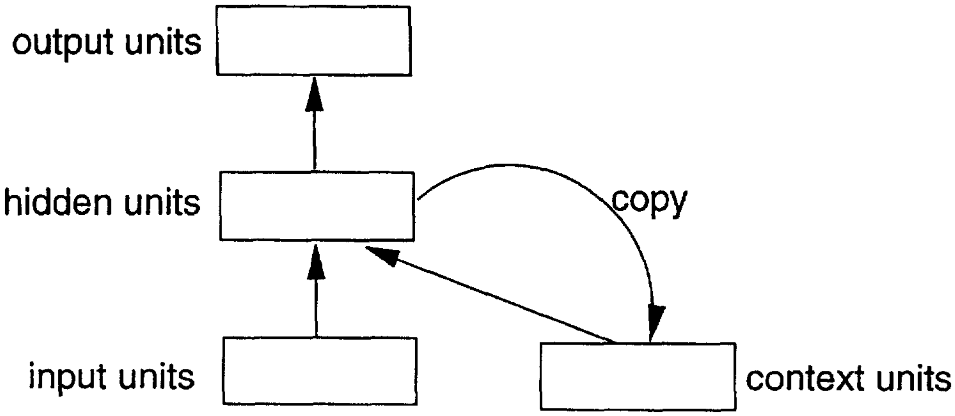

Figure 2. The Simple Recurrent Network (SRN)

used by Cleeremans and McClelland (1991) will account for many of the features of how people transfer between different domains.

Cleeremans and McClelland (1991) attempted to model learning as a finite state grammar with an architecture suggested by Elman (1990), called a Simple Recurrent Network (SRN—see Figure 2). The purpose of the SRN is to learn to predict what element should come next in a temporal sequence of elements by means of a learning procedure that is completely local in time. The current element in the sequence is coded by the input units. Activation is passed from the input units and context units through the hidden units to the output units, which attempt to predict the next element in the sequence. At the next step, the input units code the next input, the pattern of activation across the hidden units is directly copied to the context units, and the procedure is repeated in order to predict the next element. At each stage weights are changed using the back propagation algorithm.

Cleeremans and McClelland (1991) used the SRN to model finite state grammar learning in a reaction time task. Participants saw one of six lights come on, and their task was to press as quickly as possible a corresponding button. Unbeknownst to participants, the sequence of positions lighting up followed transition probabilities given by a finite state grammar. During the course of 60,000 trials spread over 20 sessions, participants became progressively more sensitive to the constraints of this grammar: This was shown by participants becoming relatively faster on trials that were grammatical rather than nongrammatical. Cleeremans and McClelland found that with a version of the SRN, the model closely matched participants in terms of the degree to which the model became sensitive to different constraints in the grammar.

Dienes (1993) showed that the SRN could also simulate how participants learn to classify exemplars and nonexemplars of a finite state grammar. The SRN was exposed to training strings for the same number of epochs as participants. In the subsequent classification phase, test strings were applied to the network. Because the SRN had learned some of the constraints of the grammar, it predicted successive letters more successfully for grammatical rather than nongrammatical test strings. A classification response was based on the extent to which predicted and actual letters matched. The results showed that the SRN could classify new grammatical and nongrammatical strings at the same level as participants, and also misclassified strings in a similar way to participants (a property that several other models of human learning did not have—Dienes, 1992).

Cleeremans, Servan-Schreiber, and McClelland (1989) showed that an SRN could be trained to predict optimally all possible successors of each element of a finite state grammar. This was true even though optimal predictions required more than the immediate predecessor to be encoded. Thus, the activation across the hidden units came to encode the relevant aspects of whole sequences of elements to allow accurate prediction of the next element. On the first trial, the hidden units code information relevant to the first element; so on the second trial when these activations are copied to the context units, the hidden units can come to code information about the previous two elements; and so on, for an indefinite number of elements. Note that this occurs in a procedure that is local in time, in that the complete sequence of elements does not need to be explicitly encoded at once.

Cleeremans showed how with extended training the representations that the network develops can, under some conditions, be very close to the abstract representation of the finite-state grammar: Each state of the finite state grammar becomes encoded by a pattern (more precisely, by a set of very similar patterns) of activation over the hidden units of the network. Cleeremans argued that although this representation is abstract, in that it has coded relevant underlying dimensions, it is not abstract in that it is intimately bound to the perceptual domain in which the SRN was trained. Cleeremans showed how the SRN could readily transfer to a new task in the same domain where the correct output simply required a different mapping from the hidden units to the output units. But how could the SRN be used to model transfer between different domains, where the input and output for one domain can be arbitrarily different from that of another?

III. A CONNECTIONIST MODEL OF TRANSFER

Typically, participants in transfer conditions have been asked to listen to sequences of, for example, spoken syllables but are then asked to make grammaticality judgements of sequences of, for example, graphic symbols (Altmann et al., 1995). Any system (be it computational or human) confronted with such a test phase *seems* to be in a Catch 22 situation: How can it classify the test items as grammatical or nongrammatical without knowing the mapping between the domains? On the other hand, how can it determine the mapping without knowing which items are grammatical? We will presently see how a network can bootstrap itself into the solution.

A simple adjustment to a standard SRN is to add an extra hidden layer (we call this the hidden encoding layer), as shown in Figure 3. One can think of the function of this layer as being to provide an abstract recoding of the domain-dependent coding provided by the input layer. Of course, when people perceive stimuli they also must successively recode the original input. In recognition of the fact that we may wish to consider transfer between two perceptually non-overlapping domains, the input layer has been arbitrarily drawn as two sets of units: One set codes the first domain (the D1 input units) and the other set codes the

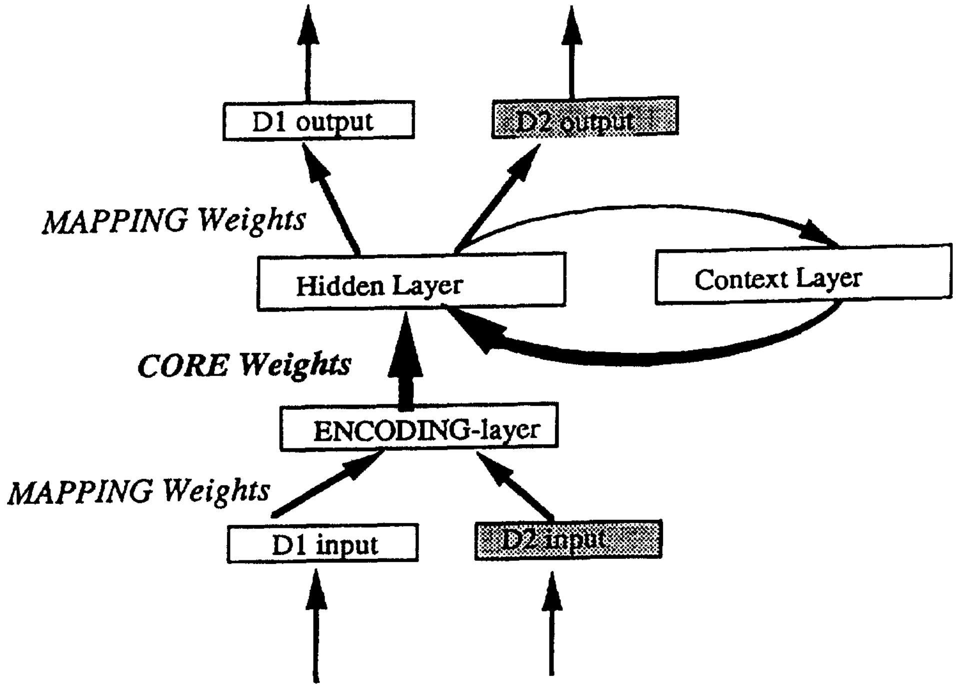

Figure 3. Modification of the SRN to enable transfer between different domains (D1 and D2).

second domain (the D2 input units). Similarly, the output layer can be considered to be one set of D1 output units and a different non-overlapping set of D2 output units.

Training

The model can be trained and tested in any one domain just like a normal SRN (Dienes, 1993; Elman, 1990; Cleeremans et al., 1989). Initially, all weights in the network are given small random values by sampling from a uniform distribution with a lower limit of -0.5 and an upper limit of +0.5. Let us say the first domain the model is trained on is melodies (Altmann et al., 1995, Experiments 1 and 2), where each melody can be made from combining and repeating any of *m* tones. The input and output D1 layers would then consist of (*m*+2) units each, a unit for each tone plus a unit to code the start of a melody and a unit to code the end of a melody. Note that the units do not represent any ordinal relations between the notes; because the codings for the different notes are orthogonal, the notes are all encoded as equally dissimilar to each other. At the beginning of each melody, the "start" D1 input unit is activated (activation set to 0.9, activation of all other units set to 0.1), and the activation of all units in the context layer is set to 0.5. This pattern of activation is successively recoded by the hidden layers in order to predict the first tone of the melody at the D1 output units (target activation of the unit coding that tone is set to 0.9, the targets for all other units is set to 0.1). Weights are adjusted by backpropagation (with momentum fixed at 0.9, and biases are also adjusted for all output and hidden units). Then the first tone is applied to the

D1 input units in order to predict the second tone, and so on. The network is trained on all the melodies participants, were exposed to for a number of epochs, ideally the same number as participants received.

In a subsequent classification phase, test strings are applied to the network (weights may or may not be changed in the test phase). If the grammar has been at least partially learned, then the network should more successfully predict successive tones of grammatical rather than nongrammatical melodies when they are applied to D1. Thus, a classification response can be based on the extent to which predicted and actual tones match. Specifically, the procedure used by Dienes (1992, 1993) was employed. The cosine was found of the angle between the vector of the actual tones in each position of the melody and the vector of predicted tones in each position (if the melody was n tones long, then the predicted and actual vectors would each have length $(n+2)*(m+2)$, with the beginning and end of each melody being coded). The cosine, C, was converted into a probability of saying "grammatical" by the equation

$$p(\text{“}g\text{”}) = 1/(1 + e^{-\text{kC - T}})$$

where the parameters k and T were fixed according to the procedure of Dienes (1992). That is, the value of T was set to produce equal numbers of "grammatical" and "nongrammatical" responses over the whole test phase, and k was set so that the predicted proportion of times the network would classify strings in error twice was 5% higher than the predicted proportion of times the network would classify a string first correctly and then in error (this is a robust property of participants' data; see Dienes, 1992; and Reber, 1989; for details). The logistic equation for $p(\text{“}g\text{”})$ can be regarded as an example of the Luce choice rule; the logistic has frequently been used for determining classification responses in connectionist networks (Gluck & Bower, 1988; McClelland & Elman, 1986).

Testing in a New Domain

Now let us say the network is tested on a new domain, for example letter strings. The "core weights" (see Figure 3) are frozen, and only the D2 input and output mapping weights are changed. The freezing could be seen as an approximation to an adaptive learning rate procedure in which the learning rate shrinks as the domain is learned (Jacobs, 1988); or as part of a more complex process by which weights can be frozen if the system registers a high likelihood of unlearning useful knowledge (cf. Murre, 1992). This is discussed further in the General Discussion. The D2 mapping weights start at arbitrary random values. The "start" is activated in the D2 input units, and the network attempts to predict the first letter. Backpropagation changes the mapping weights, and the network then attempts to predict the second letter given the first, and so on. By the time the network has reached the end of the string the mapping weights have changed, so the network can iterate around the string a number of times before moving on to the next test string to make use of mapping information gained from each pass around the string. The number of iterations, I, is a free parameter in the model. As in the same-domain case, the network will classify a string as grammatical if on the last iteration around the string it can predict successive letters in the string well. Classification is tested in the same way as.the same-domain case. The vector of predicted letters is based only on the final iteration around the string.

Because the core weights implicitly encode the structure of the new domain (because the new domain has the same underlying structure as the old domain), the network just needs to learn the mapping to and from the abstract encodings formed by the SRN in order to produce transfer. But how can it determine the mapping through the noise of the nongrammatical items? In fact, the stimuli that have been used to investigate transfer in humans have included nongrammatical sequences that were nongrammatical in only one or two locations. That is, the majority of the transitions in these nongrammatical stimuli were nonetheless grammatical. Consequently, in terms of the network, the noise introduced to the mapping weights by the nongrammatical transitions is small in comparison to the signal produced by the grammatical transitions. Thus, perhaps paradoxically, the system can ignore the noise in determining the mapping weights, but use the noise in classifying the nongrammatical exemplars as nongrammatical.

Parameters

To summarize, the free parameters in the model were number of epochs, the learning rate, the number of units in the hidden layer (the number of units in the encoding layer was set to be the same), and the number of iterations through each test string.

IV. SIMULATING THE HUMAN DATA

The main representational issues that the empirical literature on artificial grammar learning has focused on is whether knowledge of artificial grammars consists of knowledge of abstract rules, of specific exemplars from the training stage, or of bigrams and trigrams. Initially, we will determine whether the model can produce transfer to about the same degree as demonstrated by people. Then, we will consider how the model fares in accounting for the empirical data used to justify the different theoretical stances about the nature of the underlying representation. In all cases below, a preset alpha level of .05 was used for significance testing.

1. The Amount of Transfer to a Different Domain

Relative to Performance in the Same Domain

The Human Data. In Experiments 1 and 2 of Altmann et al. (1995), a grammar like that in Figure 1 was used to generate simple melodies or sequences of letters. The letters M, T, V, R, and X were mapped onto the tones c, d, g, e, and a. participants, were asked to memorise 20 grammatical stimuli—half the participants memorized strings of letters and the other half memorized melodies. All participants then classified new test letter strings and new test melodies without feedback. Finally, control groups were run who were trained on randomly generated strings, or who only received the test phase. Relative to controls, classification was significantly improved for participants tested in the same modality as learn-

TABLE 1
RESULTS FROM ALTMANN, DIENES, & GOODE (1995)

	Controls	Same domain	Different domain
Experiment 1	49 (3)	58 (5)	55 (4)
Experiment 3	47 (6)	—	58 (8)

Note. Scores are percentage correct classification. Standard deviations appear in parentheses.

ing (a 9% advantage). Relative to controls, classification was also significantly improved when a switch in modality had occurred (a 6% advantage), even though participants were not informed of the mapping. The advantage of same domain over transfer was also significant. Table 1 shows the means.

Experiment 3 of Altmann et al. (1995) used a similar design as Experiment 1, but with a different grammar and different stimuli. Participants initially listened to sequences of nonsense syllables. The set of nonsense syllables was played to participants four times in a different random order each time. Next participants classified sequences of unusual graphics characters. A control group received no training. After a switch in domain classification performance was significantly higher than that of controls (by 11%, see Table 1).

Behavior of the Model. The model was trained on the same training stimuli as people for the same number of epochs (i.e., two epochs for Experiment 1 and four epochs for Experiment 3). For each set of parameter values an average of 50 runs of the model was calculated, where each run used a different random seed for setting the initial weight values. Figure 4 shows the advantage of training the core weights on the same grammar (The "transfer" data) as opposed to a different grammar to that of the test stimuli (the "control" data—core weights were set to random values). The standard error for each mean in the top graph ("letters -> music") in Figure 4 is .0041. Thus, using t-tests, for each number of iterations, same domain performance is significantly greater than control performance; and for each number of iterations above 1 iteration, transfer performance is significantly higher than control performance; also same domain is significantly higher than transfer performance for less than 11 iterations ($ps < .05$ with Bonferonni correction given that there are 9 possible comparisons on the graph). The standard error for each mean in the bottom graph ("syllables -> symbols") in Figure 4 is .0066. Thus, for each iteration, same domain performance and transfer performance are individually significantly greater than control performance.

This pattern of results was relatively insensitive to learning rate over the range 0.5 to 1.5 (see Figure 5), and relatively insensitive to the number of hidden units over the range 7 to 35.

Comparing the Model's Behavior with that of People. The human data is plotted on Figures 4 and 5 to allow comparison with the model data. Although the scale on the y-axis is common to the model and the human data, the x-axis applies only to the model. Note that the model could achieve the same classification performance as participants in both the

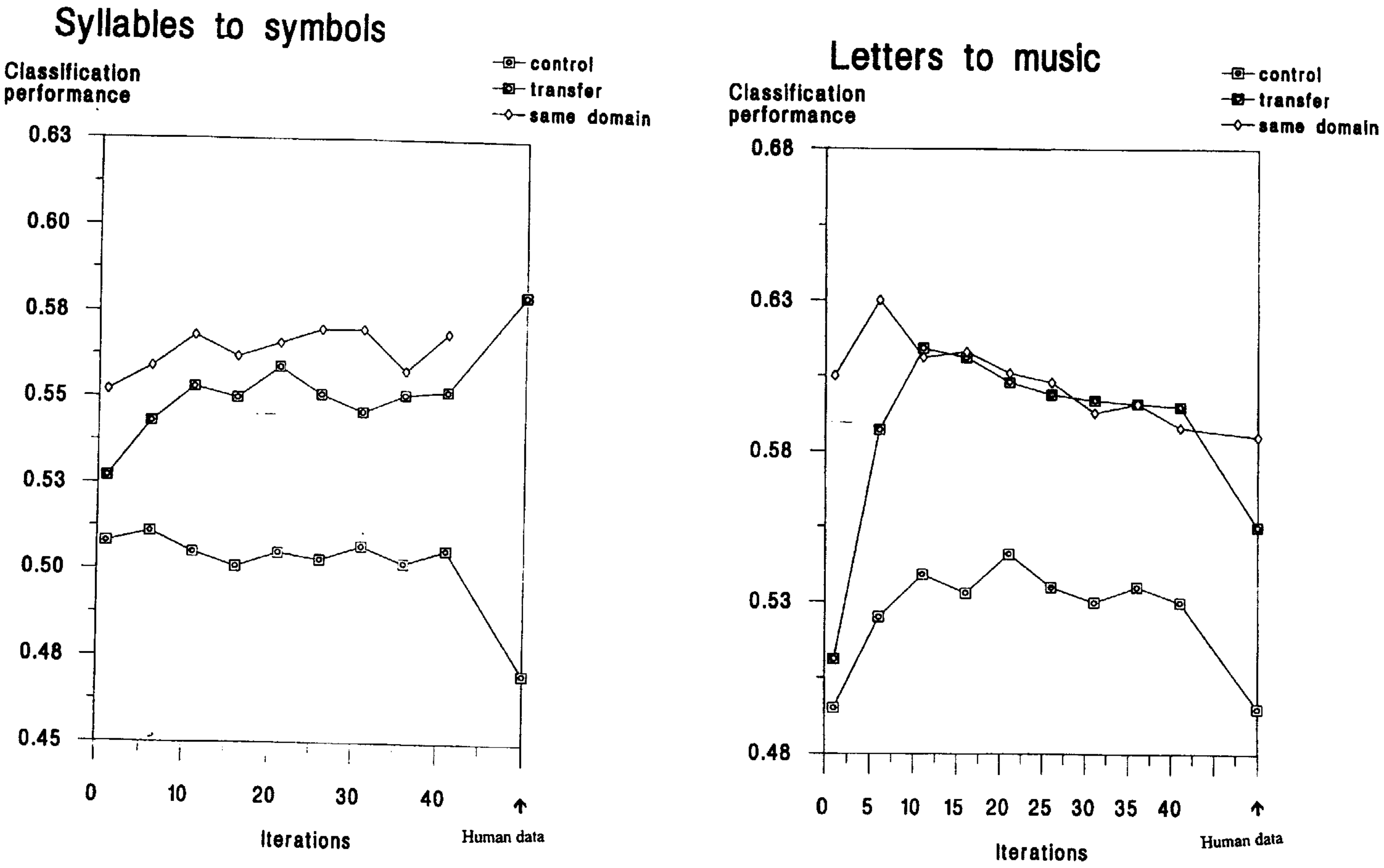

Figure 4. Simulation of Experiments 1 and 3 of Altmann et al. (1995): How performance varies with number of iterations. Learning rate was 0.9 and there 15 hidden units.

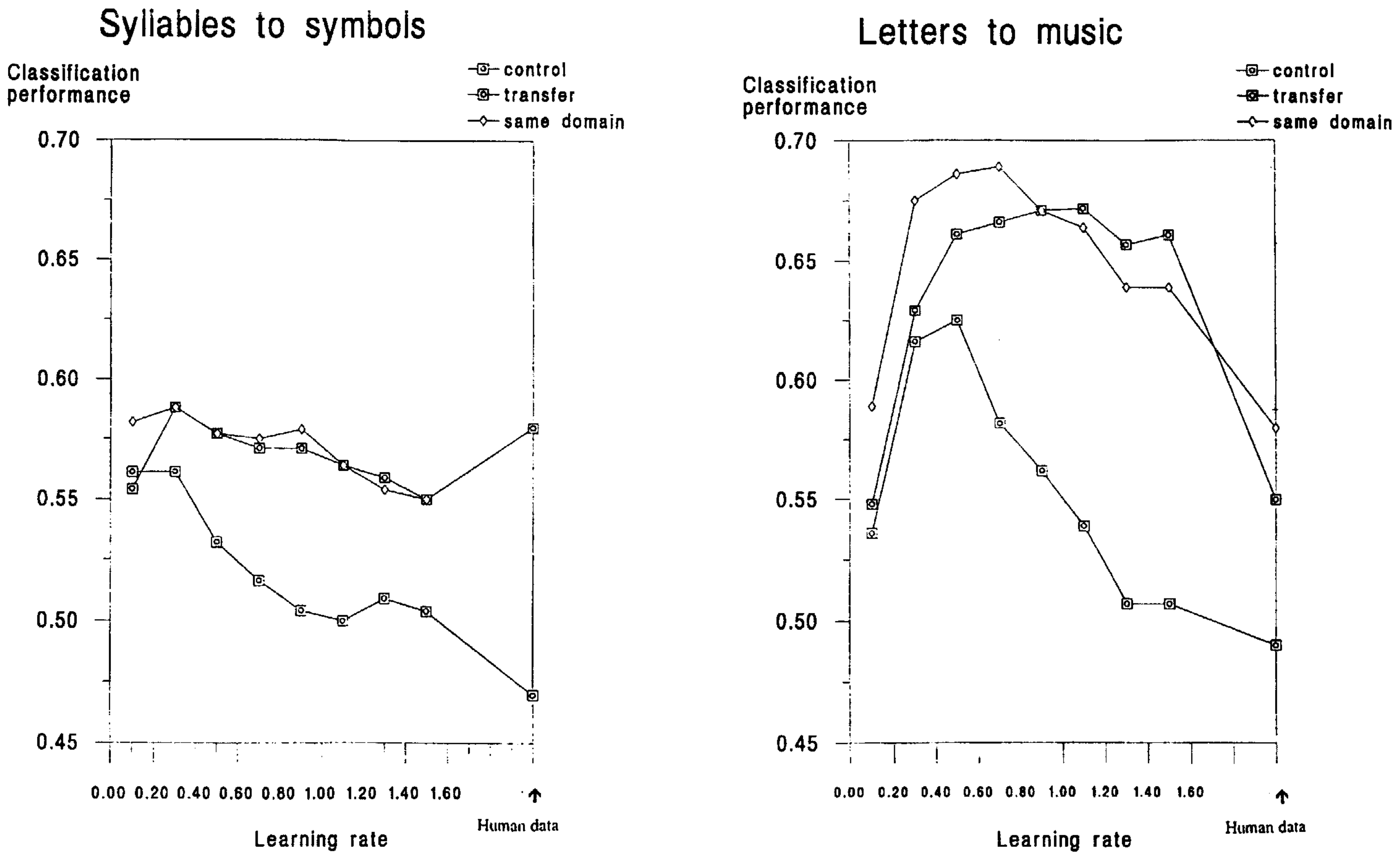

Figure 5. Simulation of Experiments 1 and 3 of Altmann et al. (1995): How performance varies with learning rate. Number of iterations was 11 there were 15 hidden units.

same domain, and transfer groups. If we take the same domain performance to define the maximum amount of cross domain transfer that could in principle be shown, then the model, like people, can perform in transfer at 70% or more of the maximum possible (see Figure 4). To provide a quantitative assessment of the fit of the model to human data, the root mean square error (RMSE) was determined separately for Experiments 1 and 3, over the parameter space defined by the dimensions of learning rate (0.1 to 1.6, steps of 0.1), and number of iterations (1 to 41, steps of 5). The error was the difference in model and human means for the different conditions (i.e., transfer, same domain, and, for Experiment 1, control); RMSE was calculated as $\sqrt{(\Sigma e^2)}/\sqrt{2}$. If the RMSE is similar in size to the variability in the human means that could be expected if the experiment were run repeatedly (as estimated by the standard error), then the model provides a good quantitative as well as qualitative fit. Number of epochs was set by the number of epochs participants were exposed to, and the number of hidden units was set at 15. For Experiment 1, the minimum RMSE was 1.85, for learning rate = 0.1 and number of iterations = 11. This is comparable to the standard error of the human means, which was 1.15. In fact, an F test indicated that the RMSE was not significantly larger than the SE, $F(3, 33) = (1.85/1.15)^2 = 2.59, p > .05$. For Experiment 3, the minimum RMSE was 1.58, for learning rate =1.1 and number of iterations = 11. This is also comparable to the standard error of the human means, which was 2.02, $F < 1$. The closeness of the RMSE to the standard error of human means indicates that any further precision in fitting a model to the means would be superfluous and effectively just fitting noise.

2. Does the Knowledge of an Artificial Grammar Consist of Knowledge of Specific Training Exemplars?

Effects of Similarity to Specific Training Exemplars

The Human Data. Brooks (1978) initially argued that people could learn concepts like artificial grammars by memorising specific exemplars and determining the similarity of test items to the stored exemplars. Brooks and Vokey (1991) argued that, just likė same-domain performance, transfer could be based on memory for specific training strings. participants could make "abstract analogies" between a test string and a specific training string. For example, MXVVVM and BDCCCB can be seen as similar because of the abstract correspondence of the relations between letters within each string. In this example, the second letter is different to the first, the third letter is repeated two times, and the final letter is the same as the first. In effect, abstract analogy is the process of comparing the repetition structure of two sequences.

In order to test this, Brooks and Vokey manipulated similarity of the test strings to the training strings in a way that was orthogonal to grammaticality as defined by a finite state grammar. "Near" test strings were only one letter different to a training string and "far" test strings were at least two letters different to any training string. They found that there was both a significant effect of grammaticality (the difference between grammatical and nongrammatical strings) and similarity (the difference between near and far strings) in deter-

TABLE 2
RESULTS FROM BROOKS & VOKEY (1991)

	Same domain	Different domain
Grammatical		
Near	64	59
Far	45	50
Nongrammatical		
Near	42	59
Far	28	37

Note. Scores are percentage of items labelled grammatical.

mining the tendency to respond "grammatical" in both the same and different domains. Table 2 shows the mean proportion of grammatical responses to test stimuli cross classified by grammaticality and similarity.

Brooks and Vokey (1991) explained their results in terms of participants, having stored an explicit list of training exemplars. Clearly, the network would not store an explicit list of training exemplars. However, the real question is, could the network reproduce the near vs far effect? The effect may arise if the model becomes sensitive to the idiosyncratic statistical structure of the training strings, which were only a sample drawn from the set of all strings generated by the grammar. That is, the training strings will not reflect exactly the statistical structure of the grammar.

Behavior of the Model. Participants were not trained for a fixed number of epochs, but had to memorize sets of four strings at a time. As simulating memorization with the model would entail further assumptions (for example, perhaps the use of fast and slow weights) and hence added complexity, the model was simply trained for 10 epochs on the same stimuli as participants learned. Figure 6 shows the proportion of grammatical responses to the test stimuli in the same domain; note that the y-axis gives the probability of responding "grammatical" rather than percentage correct classification. The standard error of each mean in the graph is .007. The model reproduces the effects of similarity and grammaticality; both effects were significant for each iteration greater than 1. Further whenever the model showed any grammaticality effect (with more than one iteration) it also showed a specific similarity effect across variations in learning rate and numbers of hidden units. So the existence of a similarity effect with these materials is a falsifiable prediction by the model of what the human data should look like. As Cleeremans (1993) showed, the SRN acquires abstract knowledge of the training stimuli. Consequently, the similarity effect produced by the model is not because of any simple explicit memorisation of the training strings. The model is instead sensitive to the regularities of the specific sample of strings on which it has been trained.

The model also reproduces the effects of similarity and grammaticality in the different domain, as shown in Figure 7. The standard error of each mean in Figure 7 is .006; both similarity and grammaticality effects were significant for each iteration greater than 1. Although the model may well be sensitive to the repetition structure of stimuli, the effect of similarity in the different domain is not because the model simply implements the

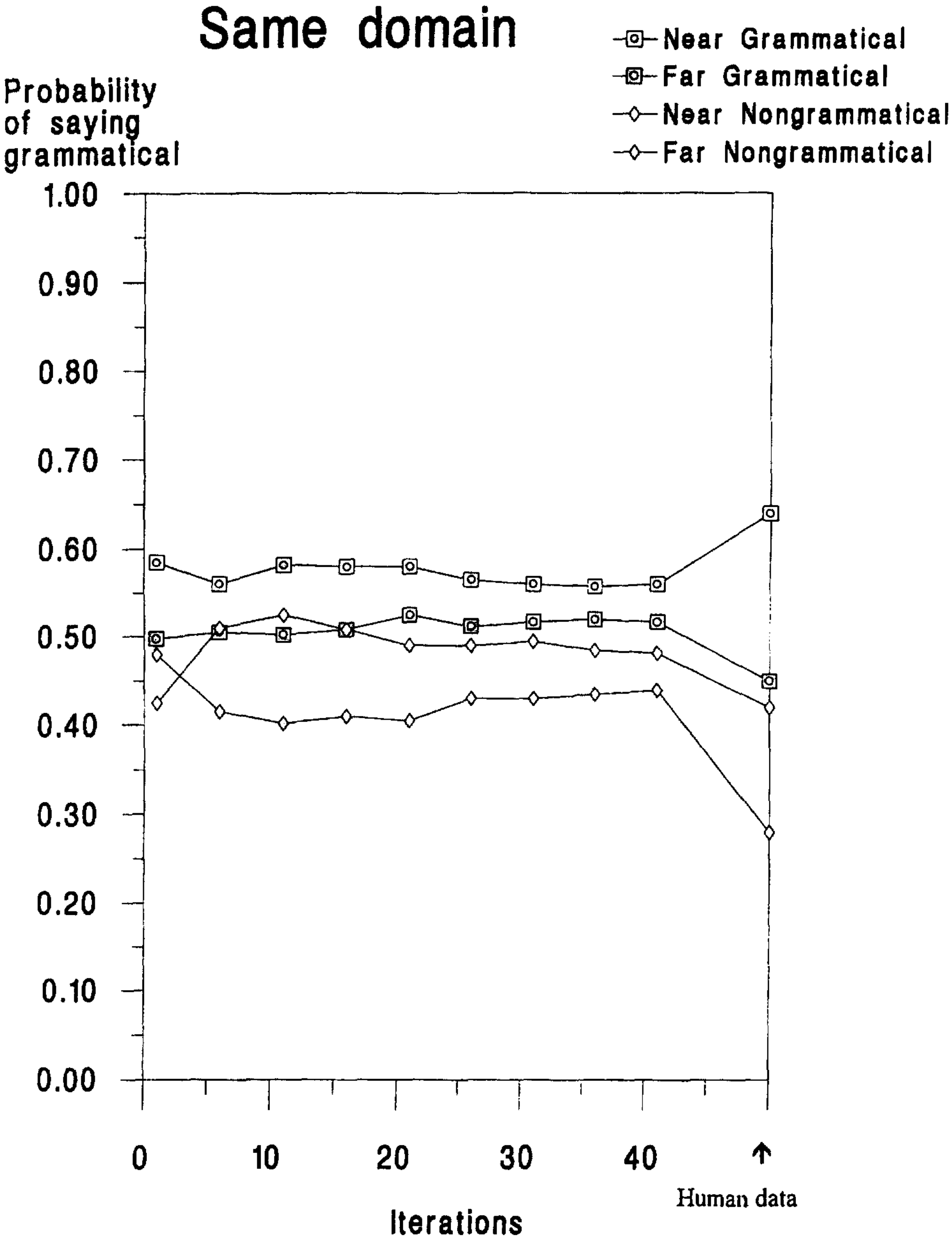

Figure 6. Simulation of Brooks & Vokey (1991) same domain. Learning rate was 0.9 and there 15 hidden units.

"abstract analogy" strategy suggested by Brooks and Vokey (1991). As we will see below, the model can correctly classify stimuli with no repetition structure.

The pattern of results as shown in Figures 6 and 7 is relatively insensitive to learning rate at least over the range 0.7 to 1.3, and to numbers of hidden units at least over the range 15 to 25.

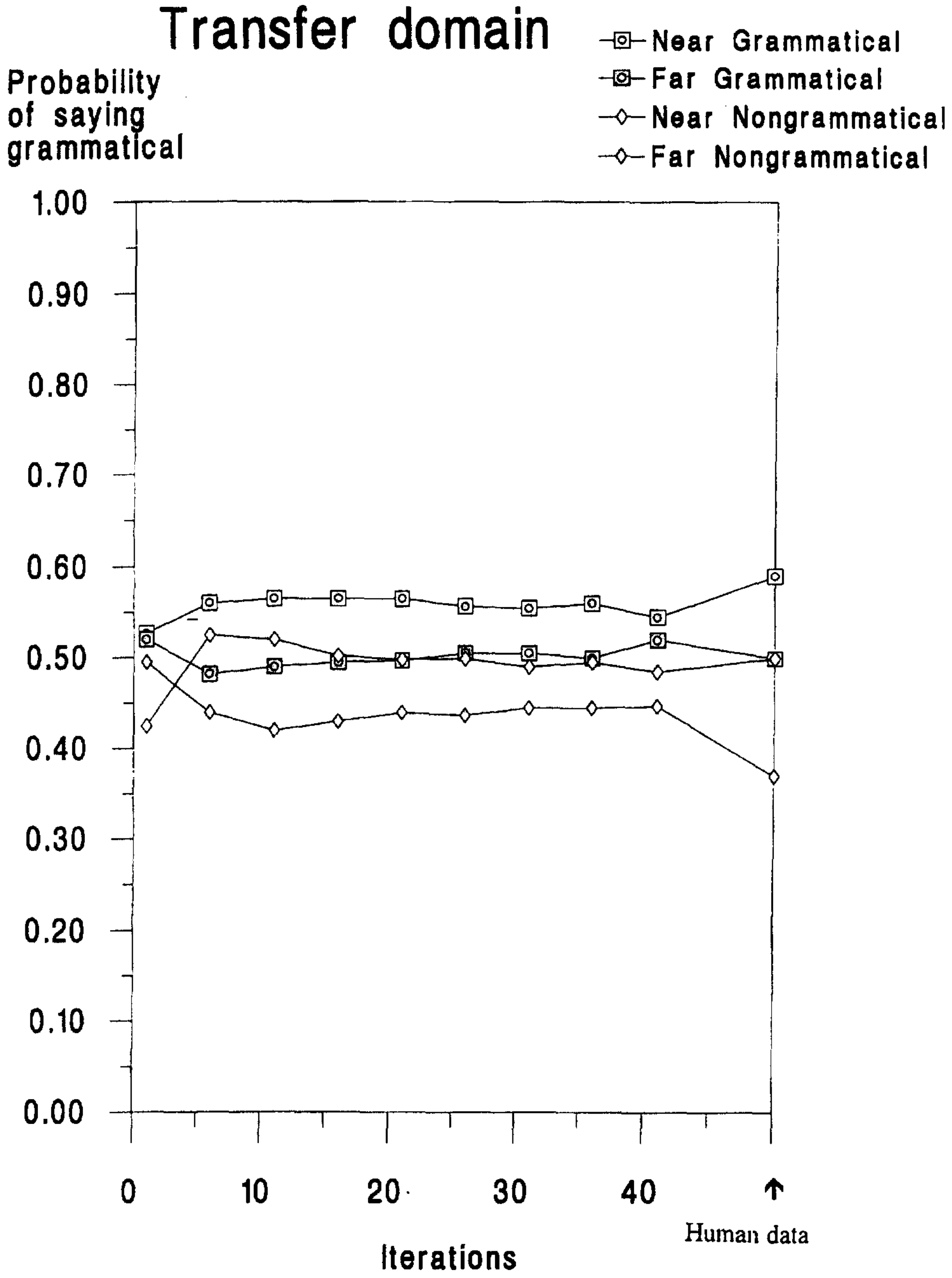

Figure 7. Simulation of Brooks & Vokey (1991) transfer domain. Learning rate was 0.9 and there wer 15 hidden units.

Comparing the Model's Behavior with that of People. Note that both model and people show effects of both grammaticality and similarity. To provide a quantitative assessment of the fit of the model to human data, the root mean square error (RMSE) was determined as before for same and different domains, over the parameter space defined by the dimensions of learning rate (0.1 to 1.6, steps of 0.1), and number of iterations (1 to 41, steps of 5) . The error was the difference in model and human means for the different conditions (i.e., near grammatical, far grammatical, near nongrammatical, far nongrammatical). Number of epochs was set at 10, and the number of hidden units was set at 15. For the same domain, the minimum RMSE was 6.05, for learning rate = 0.9 and number of iterations = 11. This is larger than the standard error of the difference in human means, which was 2.15, F(4, 34) = 7.93, $p < .01$. For the different domain, the minimum RMSE was 2.21, for learning rate =0.9 and number of iterations = 11. This is statistically indistinguishable from the standard error of the difference in human means of 2.15, F(4, 34) = 1.05. The comparison of the RMSE to the standard error of human means in the same domain case indicates that while the model reproduces the effects shown by people, there is still room for improvement in fitting the data quantitatively. Nonetheless, our results demonstrate that a model which does not store raw exemplars (Cleeremans, 1993) can be sensitive to the contingencies which give rise to the near-far effect (cf. Perruchet, 1994).

On the Information Needed to Perform Abstract Analogies

The Human Data. Whittlesea and Dorken (1993) also argued that transfer was based on memory for specific training exemplars. Instead of looking at near-far effects, as did Brooks and Vokey (1991), they looked at how focusing participants' attention on the information needed to perform abstract analogies affected transfer. Specifically, they focused participants' attention on the internal repetition structure of each training stimulus. In one condition of their experiment five (the "incidental analysis" condition), participants indicated for each letter in each string whether it was repeated elsewhere in the string or not. In the other condition ("memorization"), participants were simply asked to memorise the strings, as is typically the case in artificial grammar learning experiments. In both conditions, participants were exposed to the strings for two epochs. During the test phase, the mapping between the domains was changed on a trial-by-trial basis, so that no stable mapping between the domains could be usefully induced. That is, transfer could only be based on the repetition structure. The results are shown in Table 3. Transfer was significantly greater in the incidental analysis rather than memorisation condition. Whittlesea and Dorken concluded that abstraction is not just an automatic consequence of exposure to strings. Rather, implicit sensitivity to general structure is a by-product of whatever coding the subject uses for processing the stimuli.

TABLE 3
RESULTS FROM WHITTLESEA & DORKEN (1993; EXPERIENCE 5)

	Same domain	Different domain
Training condition:		
Incidental analysis	56	57
Memorization	59	53

Note. Scores are percentage correct classification.

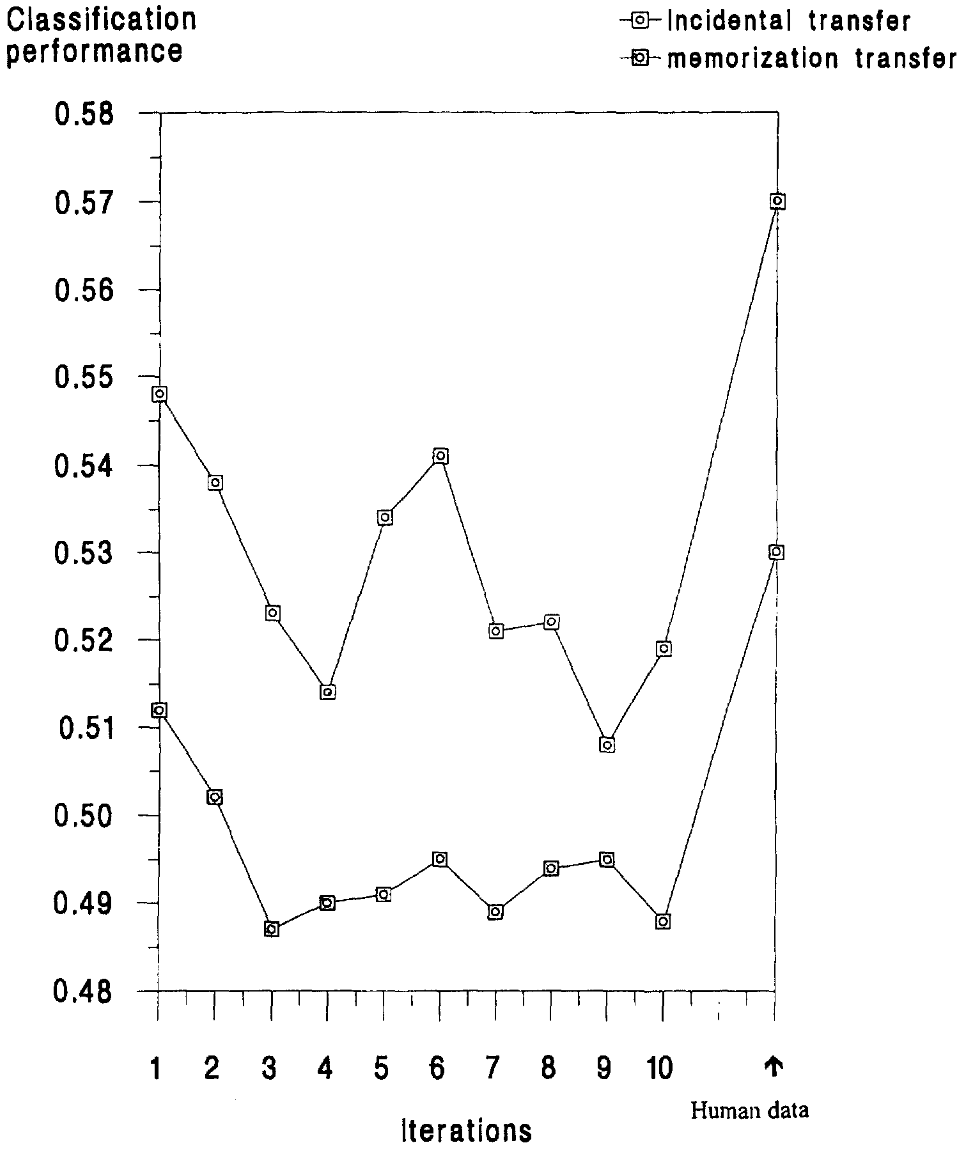

Figure 8. Simulation of Whittlesea and Dorken (1993; Experiment 5). Learning rate was 0.5 and there were 15 hidden units.

Behavior of the Model. The effect of orienting tasks was simulated in the model by assuming that the input to the network depends on the features that the participant attends to. Specifically, in the memorization condition, it was assumed, as previously in this paper, that the participant was attending to individual letters. In the incidental analysis condition it was assumed that the participant additionally attended directly to repetition structure. This was encoded by a set of six "abstract feature" units. The first unit coded whether the current letter was a repeat of the immediately preceding letter (cf. Mathews & Roussel, 1997). The second unit coded whether the current letter was a repeat of the letter two places back, and so on. Because the strings used by Whittlesea and Dorken were all seven letters long, six units were sufficient to completely capture the repetition structure. To summarize, in the memorization condition, the input and output units just coded letter features. In the incidental analysis condition, the input and output units coded both letter features and abstract features.

Figure 7 shows the results for 25 epochs. The standard error for each mean in the figure is .005; thus, for each iteration incidental transfer was significantly greater than memorization transfer. Over the range of parameter values explored ($0.1 <= lr <= 0.7$, $1 <=$ iterations $<= 31$), transfer was consistently better in the incidental analysis rather than memorization condition.

Comparing the Model's Behavior with that of People. To provide a quantitative assessment of the fit of the model to human data, the root mean square error (RMSE) was determined, over the parameter space defined by the dimensions of learning rate (0.1 to 0.7, steps of 0.1), and number of iterations (1 to 31, steps of 5). The error was the difference in model and human means for the different conditions (i.e., incidental transfer and memorization transfer). Number of epochs was set at 25, and the number of hidden units was set at 15. The minimum RMSE was 1.25, for learning rate = 0.5 and number of iterations = 11. The standard error for the human data was 1.28. The closeness of the RMSE to the standard error of human means ($F < 1$) indicates that the fit of the model was as good as the data justify.

Note that participants were only exposed to the training strings for two epochs. Although the model could show good performance after training for 25 epochs, two epochs, with one or more iterations per epoch, was not sufficient. This is discussed further in the General Discussion.

Transfer to Stimuli with No Repeated Elements

The Human Data. If grammatical and nongrammatical test strings have no repeated elements in them (either adjacently or anywhere else in the string), then they have identical repetition structures (i.e., no repetition). In this case, abstract analogy will not be able to distinguish the grammatical and nongrammatical strings. Altmann et al. (1995) showed that transfer occurred to test strings in which every element was different. Specifically, 34 of the 80 test strings used in Experiment 3 had no repetition structure, but about half of these strings were grammatical and half nongrammatical. The difference in classification performance of these strings between transfer and control participants was 9% and significant.

Behavior of the Model. Over a range of parameters, the model could consistently produce an advantage of the transfer over the control group of about 6% for the strings with no repetition structure.

During the process of classifying test stimuli, the model induces a correct mapping (or rather a close enough approximation) in the mapping weights, enabling the network to

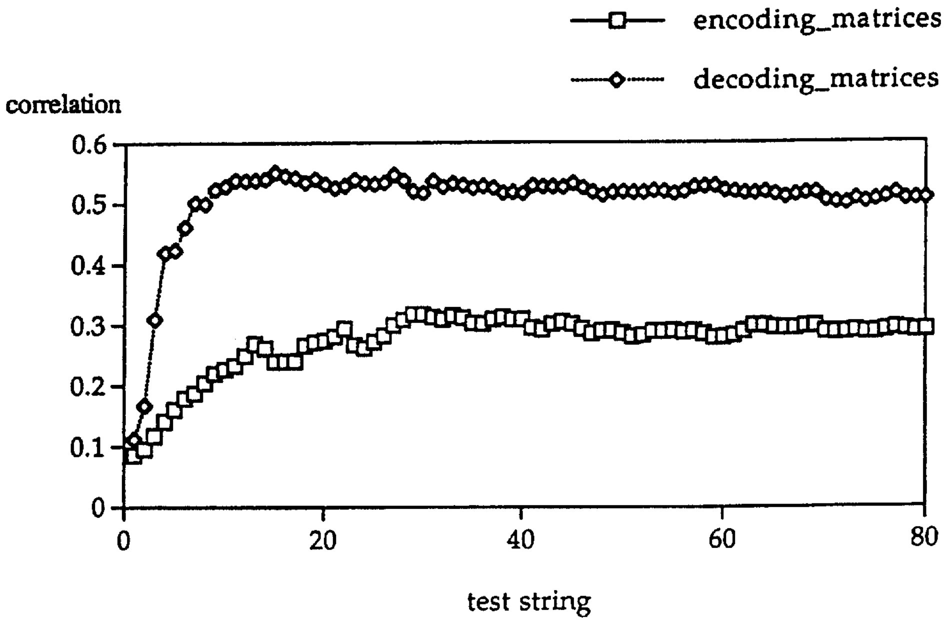

Figure 9. Correlations between mapping matrices. The encoding matrices are the matrices of weights from the input layers (D1 and D2) to the encoding layer. The decoding matrices are the matrices of weights from the hidden layer to the output layers (D1 and D2). Learning rate was 0.5, there were 6 iterations. Control performance was 52%, same domain performance was 59%, and tansfer performance was 56%.

classify stimuli with no repetition structure. Figure 9 shows how the mapping is gradually and partially induced over the first dozen test items. The rationale underlying the correlations displayed in Figure 9 is as follows. The stimuli in D2 are encoded in the same way as the stimuli in D1. In other words, stimuli which correspond according to the mapping are given the same encoding. Thus, a perfect mapping would be found when the weight matrix from D2 to the encoding layer is identical to the weight matrix from D1 to the encoding layer. This is of course just a coincidence of how we have encoded the stimuli. This is not information given to the network as the network doesn't discriminate different orderings of the units within either D1 or D2. Any form of encoding within D2 would be equivalent from the network's point of view so long as it was a linear transformation of the corresponding encoding used in D1 (because in this case there will be D1 and D2 weights matrices that converts corresponding D1 and D2 input representations into the same encoding layer representation). Thus, the Pearson correlation between the weight matrix from D1 to the encoding layer and the weight matrix from D2 to the encoding layer was calculated (called the correlation between encoding matrices in Figure 9). Also the Pearson correlation between the weight matrix from the hidden layer to D1 and the weight matrix from the hidden layer to D2 was calculated (called the correlation between decoding matrices in Figure 9). Figure 9 shows an average of 50 runs, each run starting with a different initial set of random weights. As can be seen, after about a dozen test strings, the correlations level out at moderate positive values. Although the correlations were positive averaged over runs, on any particular run the correlations need not be positive and the network could show just as good transfer. However, if the correlations were negative for the encoding matrices, they were also negative for the decoding matrices, and vice versa, in a compensatory way. That is, there was more than one mapping solution, but the most common solution found by the network was one in which the D1 and D2 matrices were positively correlated.

In summary, the partial induction of the mapping allowed the network, like people, to classify correctly strings that were composed entirely of different letters at the same level of performance as other strings.

3. Does the Knowledge of an Artificial Grammar Consist of Knowledge of Bigrams?

The Human Data. Perruchet and Pacteau (1990) originally suggested that people's knowledge of an artificial grammar could consist predominantly of knowledge of bigrams, or perhaps other n-grams of low order. Perruchet and Pacteau showed that if participants, were only exposed to bigrams, classification performance was above chance and close to levels achieved by participants, trained on whole strings. participants, are also sensitive to trigrams after a short training period (Dienes, Broadbent, & Berry, 1991), and presumably higher order n-grams after more extensive training (Servan-Schreiber & Anderson, 1990).

Gomez and Schvaneveldt (1994, Experiment 3) tested the relevance of bigram knowledge to transfer. They trained participants either only on legitimate bigrams or on whole strings (see the original paper for materials). The string participants, were exposed to 18 strings; and the bigram participants were exposed to the 17 bigrams that constituted these strings with frequencies that were in proportion to the frequencies with which the bigrams occurred in the strings. To equate the total number of letters that string and bigram participants, were exposed to, the bigram participants had a learning set of 75 bigram tokens (the 17 bigrams with their repetitions). Next, participants were tested on strings constructed from either the same letter set (same domain) or a different letter set (different domain). Classification performance is shown in Table 4. The important results were that performance in the same domain was significantly greater than chance both for participants trained on strings and those trained on bigrams, and that performance in the different domain was significantly greater than chance only for participants trained on strings. String

TABLE 4
RESULTS FROM GOMEZ & SCHAVENEVELDT (1994).

	Same domain	Different domain
Training condition:		
Strings	60 (8)	56 (7)
Bigrams	53 (6)	51 (6)

Note. Scores are percentage correct classification. Standard deviations appear in parentheses.

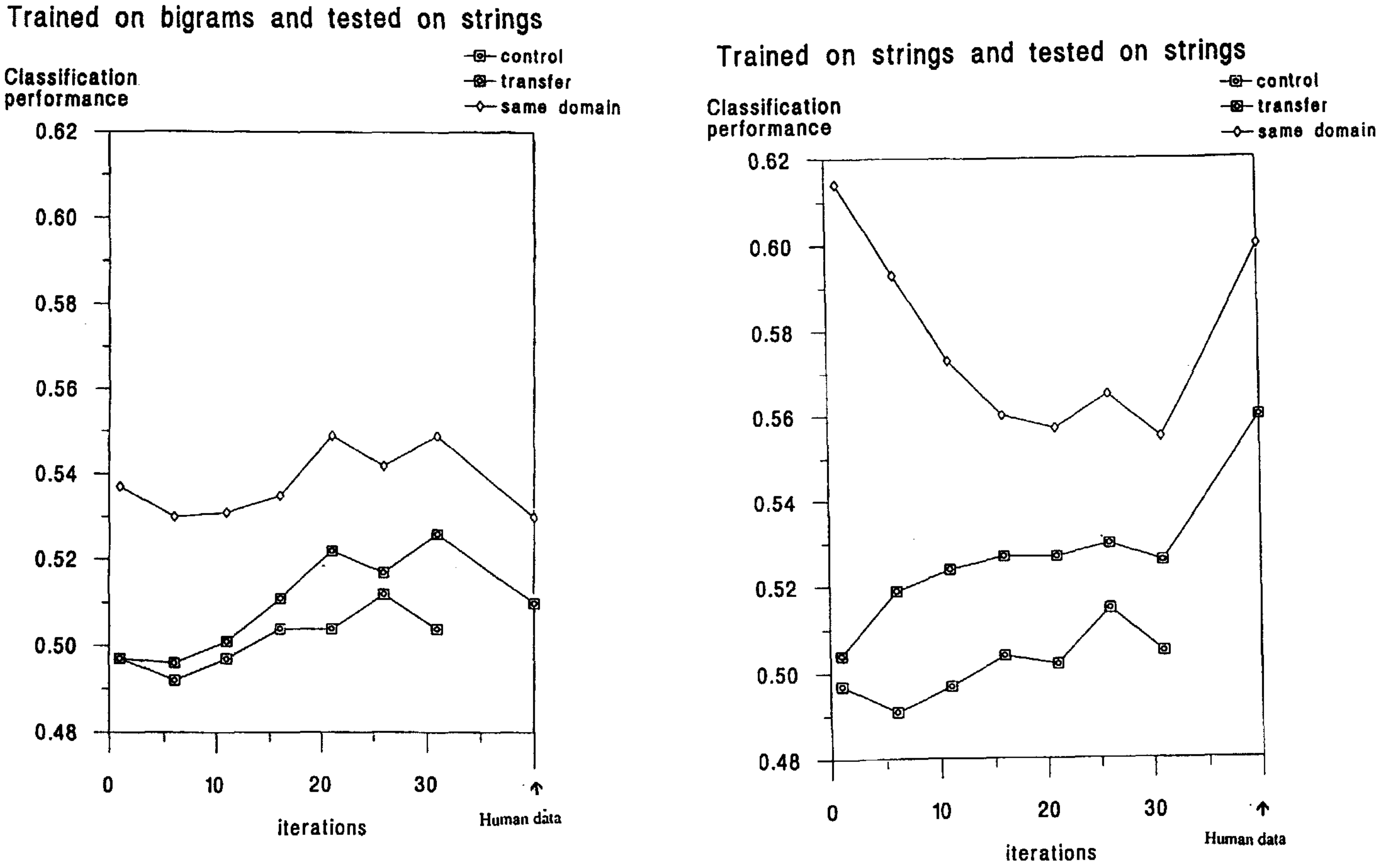

Figure 10. Simulations of Gomez & Schvaneveldt (1994; Experiment 3). Learning rate was 0.09 and there were 15 hidden units.

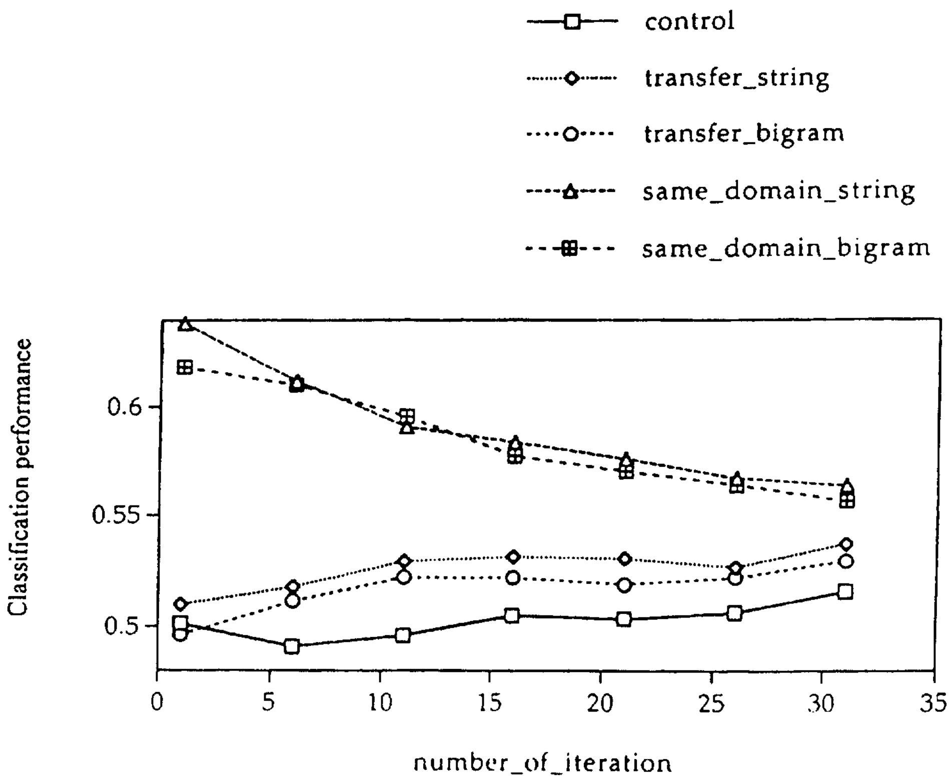

Figure 11. Simulation of training on strings and bigrams for an equal number of bigrams. The learning rate was 0.07 and there were 15 hidden units.

participants significantly outperformed bigram participants in both domains. Gomez and Schvaneveldt concluded that transfer must be based on more than just bigram knowledge.

Behavior of the Model. Gomez and Schvaneveldt trained participants, with a memorisation criterion rather than a set number of epochs. However, to simplify the modelling, the model was just trained for a set number of epochs (an epoch being 18 strings or 75 bigrams). When the network was trained on bigrams, the recurrent links were not used for either the training or test phases. The first letter of the bigram was just used to predict the second for each bigram, so the network reduced to a feedforward network.

Figure 10 shows the network's performance when it was trained with 20 epochs. The standard error of each mean was .006. Same domain performance when the network was trained on strings was uniformly good over the parameter range shown, and always significantly better than when the network was trained on bigrams. Transfer performance was also significantly better when the network was trained on strings rather than bigrams.

Notice that in Figure 10 same domain performance decreases as iterations increase, while transfer improves. This might occur because for transfer the problem is to learn the mapping weights starting from random starting values. For same domain, the mapping weights start off fairly close to optimum, and if they are trained further in the test phase, repeated exposure to the nongrammatical transitions may be detrimental.

Comparing the Model's Behavior with that of People. To provide a quantitative assessment of the fit of the model to human data, the root mean square error (RMSE) was determined, over the parameter space defined by the dimensions of learning rate (0.05 to 1.00, steps of 0.01), and number of iterations (1 to 31, steps of 5) . The error was the difference in model and human means for the different conditions (i.e., trained on strings, tested same domain; trained on strings, tested different domain; trained on bigrams, tested same domain; trained on bigrams, tested different domain). Number of epochs was set at 20, and the number of hidden units was set at 15. The minimum RMSE was 1.50, for learning rate = 0.09 and number of iterations = 6. The standard error for human classification performance is not reported in Gomez and Schvaneveldt (1994), but the maximum level of transfer in the model (less than 53%) is somewhat less than the transfer achieved by people (56%).

Redington and Chater (1995) pointed out that in the Gomez and Schvaneveldt (1994) experiment, string participants were exposed to 132 bigrams, and bigram participants to only 75 (the difference arose because Gomez and Schvaneveldt controlled for the number of letters rather than bigrams in each condition). Thus, the string participants were exposed to more bigrams than the bigram participants and this fact alone could account for at least some of the advantage of the string participants over the bigram participants in both the same and different domains. To test the importance of this issue for the model, we trained it on the same number of bigrams in both conditions. Figure 11 shows the results for when the network was trained on 30 epochs for strings and 50 epochs for bigrams, thereby approximately equating the amount of bigram exposure. Same domain performance was virtually identical, and transfer performance was also more closely matched. The network trained on strings rather than bigrams still showed superior transfer performance (perhaps because of its ability to take into account higher-order n-grams), but the difference was never more than 1% ($0.05 \leq lr \leq 1$, $1 \leq$ iterations ≤ 31). In summary, the model predicts that when bigram exposure and same domain performance are equated in strings and bigram training conditions, transfer should also be approximately equivalent to within a percent. This prediction has yet to be tested with people.

V. GENERAL DISCUSSION

The purpose of this paper has been to explore the type of model that could apply knowledge of the structure of one domain to another in the same way as people. Simply adding an extra hidden layer to the Simple Recurrent Network introduced by Elman (1990) enabled it to transfer knowledge of a finite state grammar between two different domains; to be responsive to both the grammaticality of the test strings and their similarity to training strings; to show greater transfer when trained on whole strings rather than bigrams equated for the same number of letters; and to classify test strings with no repetition structure. In addition, using the appropriate encoding, sensitivity was also found to the cover task used during training.

One criticism of the model is that the freezing of the core weights as implemented here is a purely arbitrary way of protecting against unlearning.[2] In fact, if the weights are not frozen the network shows no transfer, effectively because of catastrophic interference (McCloskey & Cohen, 1989). The noisy input in the new domain (noisy because the mapping weights start out with random values) causes unlearning of the knowledge of the old domain. One solution is to regard the freezing as an approximation to an adaptive learning rate procedure. As the output of the network comes to more closely match the target output during training, it would be expected that the optimal learning rate would get progressively smaller in order to make progressively finer adjustments in the rates. That is, if there were some procedure that systematically reduced the learning rate to near zero by the end of training in order to ensure optimal learning, there would be independent justification of the need to freeze the weights.[3] This solution assumes that learning in people asymptotes after the few minutes normally given for training in the experiments simulated above: However, Mathews et al. (1989) found that participants' knowledge of an artificial grammar steadily improved over several hours exposure. It seems unlikely therefore that participants' really do spontaneously freeze their weights just due to the passage of a few minutes learning.

Hetherington and Seidenberg (1989) showed that interleaving the learning of different items could effectively reduce catastrophic interference. Along similar lines, McClelland, McNaughton, and O'Reilly (1995) suggested that the problem of catastrophic interference might be solved in the brain by the use of two complementary systems: One, in the neocortex, discovers the structure in an ensemble of items, and requires the interleaving of items so that subsequent items do not cause forgetting of previous items (that is, this system is susceptible to catastrophic interference); the other, in the hippocampus, allows the rapid learning of new items but does not attempt to abstract common structure, and so is not susceptible to catastrophic interference. McClelland et al. propose that the hippocampus continually reinstates new and old memories so as to integrate them into the structured neocortical memory system, thereby protecting it from interference. In this paper, we are arguing that the brain learns artificial grammars with a neural network like an SRN, which requires interleaved learning of items and is susceptible to catastrophic interference. Following the logic of the McClelland et al. proposal, it may be that exposure to the new domain does not produce interference in the core weights because the hippocampus continually provides the network with examples of strings or string fragments from the old domain ensuring that the core weights keep their weight configuration. This idea suggests that the core weights would be protected during testing particularly when the relevance of the old domain is made clear to participants; this cueing of the old domain might ensure that the hippocampus reinstates relevant old memories.[4]

In order for the model to fit the human data, generally more than one iteration was needed. This makes intuitive sense in terms of the logic the model—the mapping weights may have changed after the first iteration, and the information about mapping gained from the end of the string can be applied to the beginning of the string by iterating more than once. But some may criticize the lack of an exact correspondence between human trials and iterations. However, it is unclear whether a simple correspondence exists between number of iterations in the model and any observable measure of iterative processing in human participants; for example, a participant may make just a single visual pass through a sequence but then, subsequently, iteratively process the ensuing representation. Future tachistoscopic studies with masked stimuli could bear on this issue.

The model reported in this paper addresses claims made in the artificial grammar learning literature about the nature of the acquired knowledge. Reber (1989) regarded the knowledge acquired by people in typical artificial grammar learning experiments as having two key properties. First, he argued that the knowledge was *implicit* because participants, found it very difficult to report what the rules of the grammar were. This claim has aroused controversy (see Dienes & Berry, 1997; Shanks & St John, 1994; for reviews) and the answer depends on how implicit is defined. This paper has not directly addressed the question of whether the transfer of knowledge of artificial grammars across domains is really implicit. However, the use of a connectionist network to model the process has implications for this question. Dienes and Perner (1996) argued that knowledge in a connectionist network is implicit in that it is not generally represented by the network as an object of belief. Similarly, participants., lack metaknowledge about their knowledge of artificial grammars (Dienes, Altmann, Kwan, & Goode, 1995). In the case of transfer, participants, can believe that they are literally guessing and still show substantial transfer (Dienes & Altmann, 1997).

Reber (1989) claimed that the knowledge people acquire about artificial grammars is not only implicit but also typically abstract, and we will consider this claim next. Knowledge can be abstract in different ways, and one needs to be clear about which sense of abstractness different data are relevant to (see Mathews, 1990). According to one sense of abstract, knowledge of an artificial grammar is abstract in that it is not strongly tied to specific perceptual features. This paper has shown how in this sense knowledge in a connectionist model can be abstract. Previous models of artificial grammar learning had difficulty modelling transfer because previous models did not code features in a sufficiently abstract way. Most models coded letters as the basic features (e.g., the competitive chunking model of Servan-Schreiber & Anderson, 1990), or letters in a particular position as features (e.g., the autoassociators of Dienes, 1992). The current model can go beyond the simple perceptual features given by the experimenter because the hidden layers allow recoding of the experimentally specified features.

According to another sense of abstract, knowledge is abstract if its function is to indicate the relations between features in the input (e.g., training strings) as a whole. For example, abstract knowledge of a set of strings is knowledge that explicitly represents collective or common properties of the strings. Such knowledge can then apply to strings or parts of strings that were not presented during training. For example, the following knowledge is abstract in this sense: "In the training set, T can immediately repeat itself after an M", "X cannot start", etc. At first, it may seem that if participants can classify strings that they were not trained on, then their knowledge must be abstract in this sense. However, participants' knowledge could generalize from the training strings to the test strings not because participants have actively induced a more abstract representation (that is, a representation whose function is to indicate commonalities in the training strings), but because participants can only remember part of each training string (Perruchet & Pacteau, 1990; Mathews, 1991). For example, participants may predominantly remember which bigrams or higher order fragments are permitted by the grammar, and this would allow classification of novel strings. The SRN starts by simply learning the bigrams that occur commonly in the training strings, and then building up to trigrams and then progressively higher-order n-grams (Cleeremans, 1993). In this way, the representations formed by the SRN might be thought similar to the fragmentary knowledge postulated by Perruchet and Pacteau (1991) and others. However, in the case of the SRN, knowledge of bigrams and other higher-order n-grams is abstract in that it is the function of the SRN to discern the commonalities in the training strings; it is not the case that bigrams and other *n*-grams are learned because the SRN has "forgotten" other details. Thus, the *n*-gram knowledge of the model can be seen as abstract and consistent with Reber's claims.

Brooks (1978) was the first to point out that participants could memorize training strings and then classify test strings on the basis of their similarity to the training strings, and thus classification need not be based on abstract knowledge. Evidence for this claim was provided by Vokey and Brooks' (1992) who demonstrated that people showed similarity effects as well as grammaticality effects. The model presented in this paper showed how both similarity and grammaticality effects could be produced by a model that induced abstract knowledge.

The success of the SRN in modelling these data illustrates how abstract—i.e., not perceptually bound—knowledge of an artificial grammar can be understood simply in terms of sensitivity to statistical structure. One criticism of connectionist models generally has been that because they are simply sensitive to statistical structure defined over the features of a particular domain, their knowledge is highly inflexible and domain-dependent (Clark & Karmiloff-Smith, 1993). Hinton (1990) had shown how different domains (different families) with the same structure (family trees) could come to use the same representations coded in the hidden layer when both domains were trained simultaneously. This paper has demonstrated how the knowledge embedded in a connectionist model trained in one domain can be later transferred to a different domain with an arbitrarily different front-end content (just so long as there is a linear mapping between the front-end contents of the different domains). In this sense, the model appears as flexible as participants in manipulating its implicit knowledge of artificial grammars in a domain-independent way. However, consistent with the claims of Clark and Karmiloff-Smith, there is a sense in which the knowledge in the model is remarkably inflexible. Despite the fact that it effectively induces a mapping between the domains, the mapping between D1 and D2 units (see Figure 3) is not explicitly represented anywhere in the system, and so the model would not be able to report what the mapping was between the two domains without a further system to redescribe the knowledge in the model. In fact, people also appear unable to report the mapping between the two domains (Dienes & Altmann, 1993), demonstrating a similar sort of inflexibility as the model.

Despite these successes, one indication of a weakness in the model was in terms of the amount of training required to simulate the learning of different grammars in different experiments. For example, Altmann et al. (1995, Experiments 1 and 2) trained participants for two epochs, and the SRN could match participants' performance also after two epochs. However, Whittlesea and Dorken (1993) trained participants for two epochs, and the model required 20 epochs to match participants' performance. That is, the relative difficulty the model had with learning different grammars wasn't matched by participants' performance. One difference between the grammars that may be important is that the Altmann et al. grammar disallowed more bigrams; that is, 40% of bigrams were not allowed by the grammar. On the other hand, only 12% of bigrams were not allowed by the Whittlesea and Dorken grammar. Good performance on the Whittlesea and Dorken materials may require knowledge of higher-order structure. Cleeremans (1993) showed on a sequential reaction time task how participants can sometimes become sensitive to higher-order structure at a faster rate than the SRN does. Kruschke (1993) argued that in general backprop with sigmoidal hidden unit activation functions (as we have used) will often have difficulty learning linearly nonseparable—i.e., higher order—rather than linearly separable classifications. Thus, the problem might be addressed by using an SRN with localised rather than sigmoidal receptive fields. Future research needs to explore the determinants of the difficulty of different grammars for people and different models.

Because such a simple architecture as we have investigated can produce transfer between different domains, the question arises as to how general the transfer phenomenon might be in human psychology. Future research could explore the possibilities of using architectures like the one we have used here for modelling other examples of intermodal integration: For instance, the mapping between the auditory and visual inputs that gives rise to the McGurk effect (seeing the lips making one sound, hearing another, and perceiving a third; McGurk & MacDonald, 1976); the mapping between the visual and kinaesthetic senses that give rise to infant intermodal co-ordination (Meltzoff, 1981); or even the mapping between orthography and phonology that must be acquired during reading acquisition (see Altmann et al., 1995; Altmann, 1996, for further discussion of the relevance of intermodal mapping to issues in language acquisition).

Acknowledgment: Preparation of this article was supported by a Medical Research Council grant to the first two authors. We would like to thank Luke Boucher, Axel Cleeremans, Mark Seidenberg, and Mark St John for valuable suggestions.

NOTES

[1] Indeed, Hofstadter (1985) regarded the problem of how analogies are formed as the fundamental problem in cognitive science. This paper will address one aspect of the problem of forming an analogy; namely, how a mapping could be formed between different domains already divided into corresponding parts with an unknown one-to-one mapping.

[2] In this regard, we note that Cathy Harris (personal communication, March, 1997) has observed, when replicating Elman's simulations, that once the SRN is trained on one vocabulary, it can acquire a second vocabulary set considerably faster than if it had not previously learned the first vocabulary set (as measured in numbers of cycles to reach criterion; criterial learning was achieved after only around a quarter of the cycles required to achieve it with the first vocabulary set). Harris did not adjust the learning rate as we have done here, although we note also that there are two important differences between our own simulations and Harris's: first, our simulations included ungrammatical items at test (we did not measure learning savings), and

second, Harris's simulations employed considerably larger vocabulary sets and considerably more learning cycles, which may have led to the hidden state space being partitioned in a way that was resistant to the kind of 'unlearning' that we prevented here through adjustment of the learning rate.

3. We found that use of Jacob's (1988) adaptive learning rate procedure did not lead to overall greater learning, nor shrink the weights sufficiently to allow transfer. Another possible type of adaptive learning rate procedure is to reduce the learning rate as the correlation between output and target activations increases. A simple implementation of this idea is a rule of the form $lr_{t+1} = lr_t\, k_1(k_2\text{-AvCor})$, where lr_t is the learning rate at time t, AvCor is the average correlation between target and output activations over some number of training stimuli prior to time $t+1$, k_2 is an estimate of the maximum correlation that's expected to be achieved in the domain, and k_1 is a constant. We did not find simple parameters (based on, for example, an idealized maximum correlation of 1) which shrunk the learning rate sufficiently to allow transfer. Clearly, the maximum correlation cannot reach 1, and a mechanism is required for estimating the actual maximum that can be attained given the sequences used in training. One solution may be to update the value of k_2 dynamically as the actual correlation asymptotes (such that k_2 becomes equal to AvCor).
4. This suggested role of the hippocampus in transfer does not fit with the findings of transfer in amnesic patients by Knowlton and Squire (1996). However, the patients did perform marginally worse than controls (p = .06, averaging over transfer and same domain performance). In our lab, we have recently found that seven amnesic patients performed at chance on transfer (49% correct classifications on average).

REFERENCES

Altmann, G. T. M. (1996). Accounting for parsing principles: From parsing preferences to language acquisition. In T. Inui & J. McClelland (Eds), *Attention and Performance XVI*. Cambridge, MA: MIT Press.

Altmann, G. T. M., 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.

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

Brooks, L. (1978). Nonanalytic concept formation and memory for instances. In E. Rosch & B.B. Lloyd (Eds.), *Cognition and categorization* (pp.169–211). Hillsdale, N.J.: Erlbaum.

Brooks, L. R., & Vokey, J. R. (1991). Abstract analogies and abstracted grammars: Comments on Reber (1989) and Mathews et al.. (1989). *Journal of Experimental Psychology: General, 120*, 316–323.

Clark, A., & Karmiloff-Smith, A. (1993). The cognizer's innards: A psychological and philosophical perspective on the development of thought. *Mind & Language, 8*, 487– 5– 19.

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., Servan-Schreiber, D., & McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation, 1*, 372– 381.

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, *Implicit learning: theoretical and empirical issues*. Hove: Erlbaum.

Dienes, Z., & Altmann, G. (1993). *The transfer of implicit knowledge across domains*. Paper presented at the Toronto meeting of the EPS and BBCS, July, 1993.

Dienes, Z., & Altmann, G. (1997). Transfer of implicit knowledge across domains? How implicit and how abstract? In D. Berry (Ed.), *How implicit is implicit learning?* (pp 107–123). Oxford: Oxford University Press.

Dienes, Z., Altmann, G., Kwan, L, & Goode, A. (1995) Unconscious knowledge of artificial grammars is applied strategically. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 21*, 1322–1338.

Dienes, Z., & Berry, D. (1997). Implicit learning: below the subjective threshold. *Psychonomic Bulletin and Review, 4*, 3– 23.

Dienes, Z., Broadbent, D. E., & Berry, D. C. (1991). Implicit and explicit knowledge bases in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 17*, 875–887.

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.

Druhan, B., & Mathews, R. (1989). THIYOS: A classifier system model of implicit knowledge of artificial grammars. *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Elman, J. (1990). Finding structure in time. *Cognitive Science, 14*, 179–211.

Gluck, M. A., & Bower, G. H. (1988). Evaluating an adaptive network of human learning. *Journal of Memory and Language, 27*, 166–195.

Gomez, R. L., & Schvaneveldt, R. W. (1994). What is learned from artificial grammars? Transfer tests of simple association. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 20*, 396–410.

Hetherington, P. A., & Seidenberg, M. S. (1989). Is there "catastrophic interference" in connectionist networks? *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society* (pp 26–33). Hillsdale, NJ: Erlbaum.

Hinton, G. E. (1990). Mapping part– whole hierarchies into connectionist networks. *Artificial Intelligence, 46*, 47–75.

Hofstadter, D. R. (1985). *Metamagical themas: Questing for the essence of mind and pattern*. Middlesex: Penguin.

Jacobs, R. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks, 1*, 295–308.

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

Manza, L., & Reber, A. S. (1997). Representation of tacit knowledge: Transfer across stimulus forms and modalities. In D. Berry (Ed.), *How implicit is implicit learning?* (pp. 73–106). Oxford: Oxford University Press.

Mathews, R. C. (1990). Abstractness of implicit grammar knowledge Comments on Perruchet and Pacteau's analysis of synthetic grammar learning. *Journal of Experimental Psychology: General, 119*, 412–416.

Mathews, R. C. (1991). The forgetting algorithm: How fragmentary knowledge of exemplars can abstact knowledge. *Journal of Experimental Psychology: General, 120*, 117–119.

Mathews, R. C., Buss, R. R., Stanley, W. B., Blanchard-Fields, F., Cho, J-R., & Druhan, B. (1989). The role of implicit and explicit processes in learning from examples: A synergistic effect. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*, 1083–1100.

Mathews, R. C., & Roussel, L. G. (1997). Abstractness of implicit knowledge: A cognitive evolutionary perspective. In D. Berry (Ed.), *How implicit is implicit learning?* Oxford: Oxford University Press.

McAndrews, M. P., & Moscovitch, M. (1985). Rule-based and exemplar-based classification in artificial grammar learning. *Memory & Cognition, 13*, 469–475.

McClelland, J. L., & Elman, J. (1986). Interactive processes in speech perception: The TRACE model. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing. Explorations in the microstructure of cognition* (pp 58– 121). Cambridge, MA: MIT Press.

McClelland, J. L., McNaughton, B. L., & O'Reilly. (1995). Why there are complementary learning systems in the Hippocampus and Neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review, 102*, 419–457.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In G. H. Bower (ed.), *The psychology of learning and motivation*, Vol. 24 (pp. 109–165). New York: Academic Press.

McGurk, H., & MacDonald, J. (1976). Hearing lips and seeing voices. *Nature, 264*, 746–748.

Meltzoff, A. N. (1981). Imitation, intermodel coordination and representation in early infancy. In G. Butterworth (Ed.), *Infancy and epistemology* (pp 85–114). Brighton: Harvester Press.

Murre, J. M. J. (1992). *Learning and categorization in modular neural networks*. New York: Harvester Wheatsheaf.

Perruchet, P. (1994). Learning from complex rule– governed environments: On the proper function of conscious and unconscious processes. In C. Umilta & M. Moscovitch (Eds), *Attention and performance XV: conscious and nonconscious information processing* (pp 811–835). Cambridge, MA: MIT Press.

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

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

Reber, A. S. (1969). Transfer of syntactic structures in synthetic languages. *Journal of Experimental Psychoogy, 81*, 115–119.

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

Redington, M., & Chater, N. (1995). *Commentary on Gomez and Schvaneveldt (1994).* Unpublished manuscript.

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

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., Johnstone, T., & Staggs, L. (1997). Abstraction processes in artificial grammar learning. *Quarterly Journal of Experimental Psychology, 50A*, 216–252.

Shanks, D. R., & St. John, M. F. (1994). Characteristics of dissociable human learning systems. *Behavioural and Brain Sciences, 17*, 367–448.

Vokey, J. R., & Brooks, L. R. (1992). Salience of item knowledge in learning artificial grammars. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18*, 328–344.

Whittlesea, B. W. A., & Dorken, M. D. (1993). Incidentally, things in general are particularly determined: An episodic-processing account of implicit learning. *Journal of Experimental Psychology: General, 122*, 227–248.