

# Reassessing Combinatorial Productivity Exhibited by Simple Recurrent Networks in Language Acquisition

Francis C. K. Wong<sup>a</sup>, James W. Minett<sup>b</sup>, William S-Y Wang<sup>c</sup>  
Language Engineering Laboratory, DSP and Speech Technology Laboratory,  
Department of Electronic Engineering, The Chinese University of Hong Kong.  
franciswong@cuhk.edu.hk<sup>a</sup>, jminett@ee.cuhk.edu.hk<sup>b</sup>, wsywang@ee.cuhk.edu.hk<sup>c</sup>

**Abstract**— it has long been criticized that connectionist models are inappropriate models for language acquisition since one of the important properties, the property of generalization beyond the training space, cannot be exhibited by the networks. Recently van der Velde *et al.* have discussed the issue of the combinatorial productivity, arguing that simple recurrent networks (SRNs) fail in this regard. They have attempted to show that performance of SRNs on generalization is limited to word-word association. In this paper, we report our follow-up study with two simulations demonstrating that (i) bi-gram does not play the dominant role as claimed (ii) SRNs are indeed able to exhibit combinatorial productivity when appropriately trained.

## I. INTRODUCTION

The issue of productivity, that is the ability to generalize from exemplars to novel, but correct, inputs has long been a controversial issue in the area of connectionist modelling, particularly in the modelling of language processing and language acquisition.

Early in the late 80s when trainable multi-layer feedforward networks were introduced [1, 2], Fodor and Pylyshyn [3] criticized that the crucial characteristics, *productivity* and *systematicity*, of human cognition could not be captured by those early network models. In the late 90s, Marcus argued on various occasions [4-8] that contemporary connectionist language processing models, Elman's [9] simple recurrent network (SRN) being one such, failed to generalize from sentences on which the network was trained to novel, grammatical sentences. This is contrary to rule-based learning behaviors that Marcus observed in experiments [6] on infants learning artificial languages.

In short, the above mentioned criticisms have challenged the connectionist language processing paradigm by questioning the ability of connectionist models to generalize beyond the training space. Some attempts have been made to tackle these problems by redesigning the training data and learning tasks in giving existence proof that networks are indeed able to generalize. On top of that, connectionist modeling has been probing a much deeper question of how and when generalization occurs, both in the computational model and in actual language learning scenario [10-15].

## II. COMBINATORIAL PRODUCTIVITY

More recently van der Velde *et al.* [16, 17] addressed the

issue of productivity from another perspective. The kind of productivity they focused on is what they called combinatorial productivity, summarized as follows. For a certain sentence frame, such as a simple declarative Noun-Verb-Noun (N-V-N) sentence, the set of all possible sentences is the set of all possible *combinations* of nouns and verbs. The size of this set grows with the increase in the size of the lexicon N and V. In natural languages, the nouns and verbs are open classes making the set of all possible sentences infinite in size. Human language learners exhibit *combinatorial productivity*: that is, by learning from only a fraction of the possible combinations, one can generalize his knowledge of the language to any (or at least the majority of) novel combinations of lexical items allowed by the language.

Van der Velde *et al.* [16] established a framework in which a SRN's ability to exhibit combinatorial productivity can be estimated in terms of generalization from training to testing data. The criterion for generalization is consistent with the canonical sense of generalization used in the area of soft-computing: that is, the ability for the learning device to perform a task equally well on testing data given that it has only been optimized with training data. In the context of van de Velde *et al.* [16] and of this paper, generalization refers to how well the SRNs process novel combinations of lexical items given that the network has only been trained on a fraction of the possible combinations.

Van der Velde *et al.* [16] carried out simulations trying to show that SRNs lack such property. They divided the lexical items into several disjoint groups. For example, {'boy', 'girl', 'sees'} being nouns and verbs of one group and {'dog', 'cat', 'bites'} of another group. Networks were trained on sentences generated by taking lexical items from the *same* group, such as:

- (1) 'boy-sees-girl' and
- (2) 'dog-bites-cat'

The testing sets, which were used to evaluate SRN's ability to generalize, on the other hand, were composed of sentences involving novel combinations of lexical items from *mixed* groups. Such as:

- (3) 'boy-bites-girl' and
- (4) 'dog-sees-cat'

In (3) and (4) the verbs 'bites' and 'sees' appear in a different context, with respect to neighbouring nouns, than they do in the training set. With this design of training and

testing data, van der Velde *et al.* [16] showed that although SRNs achieve near perfection in learning the training set sentences (evaluated via *grammatical prediction error* (GPE), more on this in section III.D) they fail to generalize to testing set sentences even though all lexical items appeared in the same syntactic positions as they did in the training set. They further argued that when SRNs fail to generalize the networks resort to ‘word-word association’. Their argument was based on the similarity of SRNs’ performance on testing data with the performance expected from a bi-gram model.

Van der Velde’s report on the lack of combinatorial productivity with SRNs cast a serious doubt on the fundamental learning capability of the model. The poor performance on generalization would indeed hinder the connectionist paradigm to proceed further, as it has been criticized of being incapable to go beyond toy grammar [18].

In this paper, we report our follow-up study on the issue of combinatorial productivity with two simulation experiments. Simulation I is mainly an existence proof showing that SRNs are indeed able to show some degree of generalization. More importantly, contrary to the claim in [16], SRNs outperform what could be achieved if they learned solely the bi-gram statistics available in the training data. We follow van der Velde’s construction of the training and testing data as well as the method of evaluating a network’s ability to ‘understand’ a sentence in order to establish a common ground to compare our results with the results of van der Velde *et al.* [16]. In Simulation II, extra training sentences, which still do not overlap with testing set sentences, were generated in order to provide a reasonable condition to trigger the network to generalize. Results show that SRNs’ performance on processing testing set sentences is similar to training set sentences in most syntactic positions. Implications of the results reported here to linguistic theory will also be discussed.

### III. SIMULATION I

#### A. Network Architecture

In our first simulation, we followed the method of the experiment performed in [16], the only difference being the choice of the SRN architecture. We used a SRN with one hidden layer only between the input and output layer, instead of three as in [16]. We chose to do so because it is generally acknowledged that it is more difficult for the learning algorithm to optimize connection weights of networks with more than one hidden layer [19].

Eight nouns and eight verbs together with the relative marker, ‘who’, and the end-of-sentence mark, ‘#’, were used in generating the language. A localistic 1-in-20-bit coding scheme was implemented to encode the lexicon where each lexical item was represented with an orthogonal bit vector. The two extra bits were reserved for two extra lexical items that would be used in Simulation II. Suppose a lexical item is coded as  $(0, 1, 0, \dots, 0)$ , it is fed as an input to the network by

setting the activation level of the network’s second input neuron to 1 (activated) and the others to 0 (non-activated). In doing so each neuron in the input layer is actually coding for one of the lexical items. Networks with 20 input layer neurons were used in the experiments reported in this paper.

To acquire the grammar of the target language, one has to learn at least the sequential regularities from the sample sentences. The common way of training a SRN to achieve this is via the prediction task whereby networks are trained to predict the next word given a partial sentence. Hence, an output layer with 20 neurons was employed. In doing so, the network’s output layer activation, which is a vector of real numbers, can be interpreted as an estimate of the probability distribution indicating which word(s) the network predicts to follow given a partial sentence. For the hidden layer and the context layer (in which hidden layer activation of the pervious time step is stored) 80 neurons were employed. The architecture described above is shown in Fig. 1.

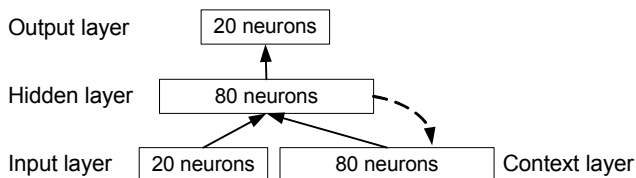


Fig. 1. SRN used in Simulations I and II. The solid lines indicate full connections between layers and the dotted line indicates the one-to-one copy-back connection from the hidden layer to the context layer. Arrows indicate directionality. The back-propagation algorithm was used to train the network. Learning rate and momentum were set to 0.1 and 0 respectively.

#### B. Training of the Networks

Networks were trained with three types of sentence, one simple and two complex (right-branching and center-embedding) sentence types, as tabulated in Table I. The training was divided into four phases of increasing complexity, in line with Elman’s ‘starting small’ training scheme [20]. In the first phase the training set consisted of simple sentences only while in the last phase the token ratio of simple-to-complex sentences was 1:4. Details of the exact number of training set sentences that were fed to the network are tabulated in Table II. The same training scheme was used in the original experiment of van der Velde *et al.* [16].

TABLE I.  
THREE TYPES OF SENTENCE USED IN SIMULATIONS I AND II

Sentence types	Frames	Natural language equivalents
Simple	N-V-N	the boy kisses the girl
Right-branching	N-V-N-who-V-N	the boy kisses the girl who chases the dog
Center-embedding	N-who-N-V-V-N	the girl who the boy kisses chases the dog

TABLE II.  
4-PHASED TRAINING SCHEME EMPLOYED IN SIMULATION I

Phase	Token and type (bracketed) ratio*	No. of sentences fed to a network
1	1 : 0 : 0 (1 : 0 : 0)	32 000
2	6 : 1 : 1 (24 : 1 : 1)	10 240
3	2 : 1 : 1 (8 : 1 : 1)	51 200
4	1 : 2 : 2 (2 : 1 : 1)	64 000

\*ratio of simple : right-branching : center-embedding

### C. Training and Testing Data

As we have mentioned earlier, the crucial difference between the training data and the testing data is that a training set sentence is generated from lexical items of the same group whereas a testing set sentence is generated from mixed groups. Here we denote  $\mathbf{n}_{ij}$  to be the  $j^{\text{th}}$  noun of group  $\mathbf{i}$ , and  $\mathbf{v}_{ij}$  to be the  $j^{\text{th}}$  verb of group  $\mathbf{i}$ .

In Simulation I, four groups of lexical items were used, each group contained two nouns and two verbs. Notice that the four sets of nouns and verbs, denoted by  $\mathbf{n}_i$  and  $\mathbf{v}_i$  ( $\mathbf{i} = 1 \dots 4$ ), were non-overlapping sets, i.e. there were altogether eight unique nouns and eight unique verbs, which was critical for the test of SRN's ability to generalize.

The training set was constructed by generating sentences composed of words from the same group. The set of right-branching training set sentences were:

Group 1:  $\{\mathbf{n}_{1a}-\mathbf{v}_{1b}-\mathbf{n}_{1c}-\mathbf{who}-\mathbf{v}_{1d}-\mathbf{n}_{1e}-\#\}$ ,

Group 2:  $\{\mathbf{n}_{2a}-\mathbf{v}_{2b}-\mathbf{n}_{2c}-\mathbf{who}-\mathbf{v}_{2d}-\mathbf{n}_{2e}-\#\}$ ,

Group 3:  $\{\mathbf{n}_{3a}-\mathbf{v}_{3b}-\mathbf{n}_{3c}-\mathbf{who}-\mathbf{v}_{3d}-\mathbf{n}_{3e}-\#\}$ ,

Group 4:  $\{\mathbf{n}_{4a}-\mathbf{v}_{4b}-\mathbf{n}_{4c}-\mathbf{who}-\mathbf{v}_{4d}-\mathbf{n}_{4e}-\#\}$

where  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e} = \{1, 2\}$

Hence 32 unique sentences ( $2^5$ , 2 different words at 5 different syntactic positions) were generated for each group. The other two types of sentences, simple and center-embedding, were generated in a similar way. The four groups of sentences were combined to form training sets with different token ratios of simple-to-complex according to the training scheme in Table II.

As for the testing sets, van der Velde *et al.* [16] designed theirs in such a way that the SRN's ability to generalize can be assessed in a systematic manner. We think it worthwhile to follow and investigate further. A testing set sentence is constructed by combining lexical items from mixed groups. The level of difficulty with respect to generalization is varied by the number of groups that are mixed. The more the number of groups the more difficult the sentence would be.

Following the convention in [16], we use  $\mathbf{N}$  to denote the number of groups that are mixed to generate a testing sentence. Examples of right-branching testing set sentences used in the simulation were:

$\mathbf{N}=1$ :  $\{\mathbf{n}_{1a}-\mathbf{v}_{1b}-\mathbf{n}_{1c}-\mathbf{who}-\mathbf{v}_{1d}-\mathbf{n}_{1e}-\#\}$ ,  
 $\{\mathbf{n}_{4a}-\mathbf{v}_{4b}-\mathbf{n}_{4c}-\mathbf{who}-\mathbf{v}_{4d}-\mathbf{n}_{4e}-\#\}$

$\mathbf{N}=2$ :  $\{\mathbf{n}_{1a}-\mathbf{v}_{3b}-\mathbf{n}_{1c}-\mathbf{who}-\mathbf{v}_{3d}-\mathbf{n}_{1e}-\#\}$ ,  
 $\{\mathbf{n}_{4a}-\mathbf{v}_{3b}-\mathbf{n}_{4c}-\mathbf{who}-\mathbf{v}_{3d}-\mathbf{n}_{4e}-\#\}$

$\mathbf{N}=3$ :  $\{\mathbf{n}_{1a}-\mathbf{v}_{3b}-\mathbf{n}_{2c}-\mathbf{who}-\mathbf{v}_{1d}-\mathbf{n}_{3e}-\#\}$ ,  
 $\{\mathbf{n}_{4a}-\mathbf{v}_{3b}-\mathbf{n}_{1c}-\mathbf{who}-\mathbf{v}_{4d}-\mathbf{n}_{3e}-\#\}$

$\mathbf{N}=4$ :  $\{\mathbf{n}_{1a}-\mathbf{v}_{3b}-\mathbf{n}_{2c}-\mathbf{who}-\mathbf{v}_{4d}-\mathbf{n}_{1e}-\#\}$ ,  
 $\{\mathbf{n}_{4a}-\mathbf{v}_{3b}-\mathbf{n}_{1c}-\mathbf{who}-\mathbf{v}_{2d}-\mathbf{n}_{4e}-\#\}$

where  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e} = \{1, 2\}$

Obviously,  $\mathbf{N}=1$  testing set is identical to the set of training sentences. The  $\mathbf{N}>1$  testing sets contain more sentences than the training set. More importantly, constructing testing set sentences this way ensures a maximum separation between lexical items from the same group since GPE is evaluated on every sentence position.

Twenty SRNs, each initialized with an independent random initial set of connection weights, were constructed and trained with streams of concatenated sentences which were randomly sampled from the training sets. The prediction task was used to train the networks. The results to be reported in the remaining parts of the paper are based on the average performance of the twenty networks.

### D. GPE Evaluation and Generalization

How well a network acquires the language can be assessed by analyzing the network's output layer activation during the processing of a sentence. Recall that words were coded in orthogonal 1-in-20 bit vectors. The activation value of an output layer neuron represents the network's estimate of the conditional probability of the word coded by that neuron to follow on the basis the partial sentence it has been exposed to. The output layer activation, which is a vector of real numbers, can be divided into 'grammatically correct' and 'grammatically incorrect' predictions, depending on the context. For example, in processing a simple sentence ( $\mathbf{N}-\mathbf{V}-\mathbf{N}-\mathbf{\#}$ ), when the first noun is fed to the network, any verb and the relative marker 'who' could be a grammatically correct continuation of the partial input since a noun can be both the start of a simple and a center-embedding sentence. Therefore, the grammatically correct prediction includes any verb and the relative marker 'who'. Likewise, at the second time step, after the verb has been fed, the correct prediction includes the set of nouns. At the third time step, it includes both 'who' and the end-of-sentence marker '#'. The grammaticality of the predictions made by the network at other syntactic positions of complex sentences was defined in a similar way. Based on this definition of grammaticality, the grammatical prediction error (GPE) [10, 16] is given by the following equation:

$$GPE = 1 - \frac{\sum \text{correct activation}}{\sum \text{correct activation} + \sum \text{incorrect activation}}$$

The value of the numerator is the sum of the activations of

those correctly activated nodes. Using a simple sentence as an example, at the first position of a sentence, the value of the numerator is the sum of activations of nine nodes, eight that code for verbs and one that codes for the relative marker. The value of the denominator is likewise the sum of activations of 18 output nodes.

Combinatorial productivity exhibited by a SRN is estimated in terms of generalization from the training set to the testing set. If the network indeed exhibits combinatorial productivity, we would expect it to attain GPE for testing set sentences as low as for training set sentences.

We take the processing of an  $N=3$  testing set sentence,  $n_{11}-v_{21}-n_{31}$ , as an example. At the position of the second noun,  $n_{31}$ , the grammatically correct continuations includes ‘who’ and ‘#’ only. If the network processes this novel combination of nouns and verbs as if it is a training set sentence such as  $n_{11}-v_{11}-n_{11}$  or  $n_{31}-v_{31}-n_{31}$ , only the nodes for ‘who’ and ‘#’ should be activated. Failing to achieve this would give high GPE value since verbs are likely to be wrongly activated due to frequently occurring  $N-V$  transitions.

### E. Results and Discussion of Simulation I

After the four phases of training, the connection weights of the networks were frozen. Testing set sentences of various sentence types and  $N$ -values were fed to the networks. GPE at different sentence positions were recorded and plotted in Fig. 2. Each data point represents the GPE in predicting the *next* lexical item given the current word, which is marked on the  $x$ -axis. Data points show the mean GPE of 20 networks where each network was evaluated with 100 sentences drawn randomly from the testing (training for  $N=1$ ) sets; the error bars indicate two standard deviations.

The GPE evaluations on testing set sentences are to be compared with some reference values:

- training set GPE ( $N=1$ ), solid thick line
- GPE on testing sets ( $N=3$ ) reported in van der Velde *et al.* [16], which we mark on Fig. 2 with diamond line markers.
- GPE obtained by a bi-gram model, solid line with circle line markers. The bi-gram model was calculated from the empirical transition probabilities (available from the training data) among the four lexical categories,  $N$ ,  $V$ , **who** and **#** during the last phase of training. The bi-gram statistics are presented in Table III.

The general trend that can be observed from Fig. 2 is that the SRN learns the training set nearly perfectly, with very low GPE, but performs worse and worse on testing sets as the complexity in terms of generalization is increased from  $N=2$  to  $N=4$ . This is consistent with the original experiment in [16]. Notice that an  $N=3$  testing set is actually as complex as an  $N=4$  testing set since in both cases the first three nouns or verbs already uniquely determine the remaining part of the sentence. This explains why the two sets of data points overlap one another.

The major claim of van der Velde *et al.* [16] was that when

a SRN fails to generalize to testing set sentences it resorts to word-word association between immediately adjacent words. They argued on the basis of their observation that testing set GPE they obtained follows the trend of a bi-gram GPE. In the remaining parts of this section we attempt to argue against this.

Firstly, the GPE of our networks attained smaller values in all sentence positions than the results reported in van der Velde *et al.* [16]. In those positions that they argued as being the most difficult to make correct grammatical prediction, such as at the final nouns of all three sentence types and at the first verb of a center-embedding sentence, the differences are substantial. The difference between our results and the results of van der Velde *et al.* can be partly explained by the difference in the choice of SRN architecture; recall that we used a SRN with one hidden layer only, while two more hidden layers were used in [16]. It is necessary to appreciate that even though connectionist models are robust in general, for example concerning the choices of learning parameters and the details of the learning algorithm, a reasonable setting of the architecture is still essential.

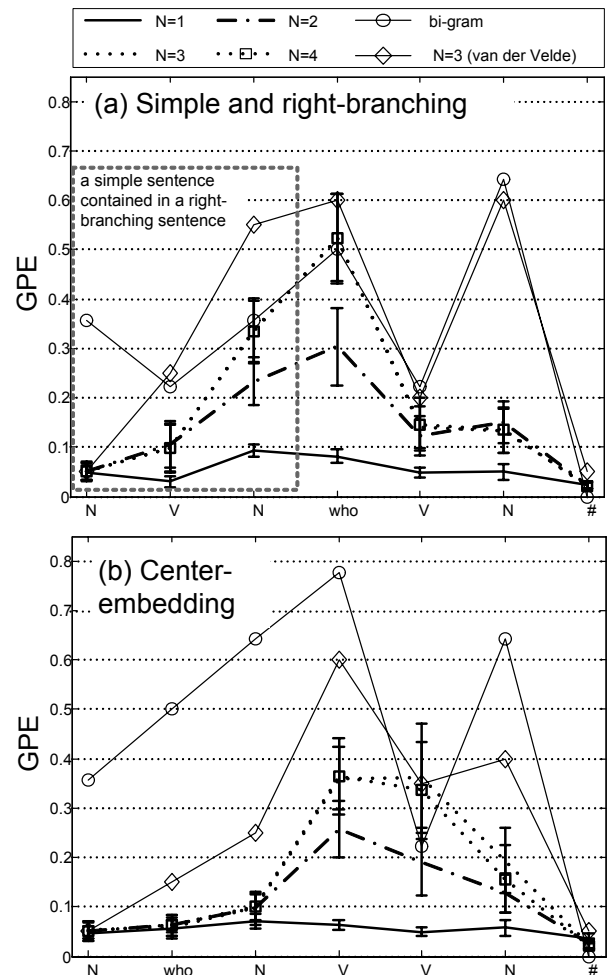


Fig. 2. Results of Sim I. Grammatical Prediction Error (GPE) in processing (a) simple, right-branching and (b) center-embedding sentences. Values for  $N=1$  to  $N=4$  are mean GPE of 20 networks each evaluated with 100 test sentences. Results that van der Velde *et al.* reported are marked on the plots with diamond line markers.

TABLE III.  
THE BI-GRAM MODEL\*

	N	V	who	#
N	0	0.357	0.286	0.357
V	0.778	0.222	0	0
who	0.5	0.5	0	0
#	1	0	0	0

\* the value of the cell in the  $i^{\text{th}}$  row  $j^{\text{th}}$  column is the probability that the words in category  $j$  follow the word in category  $i$ . Formally,  $\Pr(w_{k+1} \in C_j | w_k \in C_i)$ , where  $w_k$  and  $w_{k+1}$  are consecutive words in a sequence.

Secondly, we argue that the role of word-word association for a SRN to learn a language from exemplars was overstated in [16]. We will focus our discussion on our simulation results with testing set sentences of complexity  $N=3$  (dotted lines in Fig. 2). Among the 15 syntactic positions, excluding the end-of-sentence markers, only 4 of them have a GPE approximately equal to or larger than a bi-gram GPE. To obtain a clearer picture about the behavior of the network, we zoom in to look at the raw activation pattern of the output layer neurons during the processing of a sentence at some of the syntactic positions where high GPE are obtained.

Fig. 3 shows a snapshot of the states of a SRN in processing an  $N=3$  right-branching sentence. The bars represent the activation values of the network’s output layer neurons that indicate its prediction of what words to follow, as labeled on the  $x$ -axis. At this instance, the partial sentence that the network has seen is  $n_{12}-v_{22}-n_{32}$ .

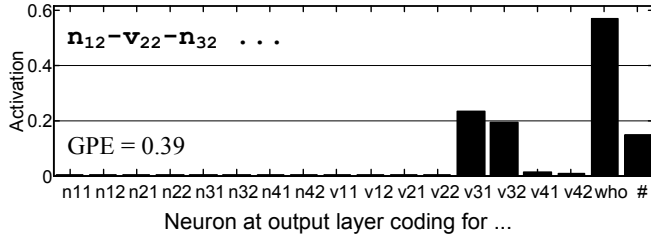


Fig. 3. Output layer activation pattern of, i.e. prediction made by, a SRN processing an  $N=3$  right-branching sentence.

This corresponds to the data points at the second noun of Fig. 2 (a) where at this sentence position the SRN gives an average GPE of about 0.35. From the activation pattern we can see that the network is incorrectly predicting  $v_{31}$  and  $v_{32}$  to follow, with probabilities of about 0.2. This indicates that the network is having some difficulty processing sentences with novel combinations of nouns and verbs from mixed groups. However, this does not mean that the SRN loses track of the sentence completely and resorts solely to the bi-gram statistics. The activation for ‘who’ at this sentence position is higher than the sum of the activation for verbs, which is clearly contrary to what might be expected from the bi-gram probabilities (the first row of Table III, where ‘who’ is expected to be the least activated).

As for another position where a bi-gram-like GPE was obtained, at the ‘who’ slot of a right-branching sentence, the activation pattern was sampled, as shown in Fig. 4.

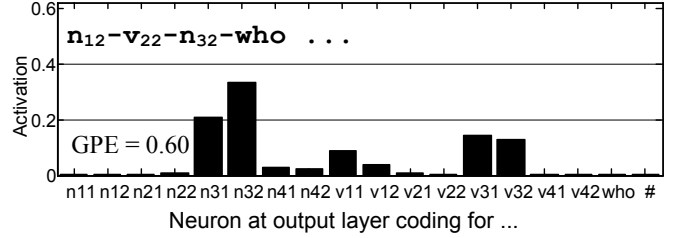


Fig. 4. Output layer activation of a SRN in processing an  $N=3$  right-branching sentence at the position ‘who’.

At this position, the grammatically correct continuation comprises verbs only. Why does the network show strong activation of the two Group 3 nouns  $n_{31}$  and  $n_{32}$ ? This cannot be accounted for by bi-gram statistics, as it would suggest all nouns and all verbs to be equally activated (the third row of Table III). Instead, we argue this is due to an artifact of the language making the SRN treat the relative marker ‘who’ as a verb. This can be understood by replacing ‘who’ with V in the frames that construct right-branching and center-embedding sentences (Table I). The replacement process would make the two sentence types identical, revealing that the relative marker ‘who’ is in fact taking a verb-like syntactic position. Hence, the network is wrongly recognizing ‘who’ as a verb and makes the prediction that a noun could also follow. To be more precise, ‘who’ is misrecognized, in this particular context, as a  $v_3$  verb and therefore  $n_3$  nouns are predicted to follow due to the preceding word  $n_{32}$ . Notice that, SRN’s sensitivity to sentence structure can still be observed as some verbs are correctly activated.

Finally, the performance degradation at the two verb positions in center-embedding sentences, for which the SRN performs poorly (Fig. 2(b)), is in fact expected [21] due to the deviation from the dominant N-V and V-N local word orders to V-V word order.

All in all, looking at the raw output layer activation of the networks reveals that the working principle of SRNs does not seem to be dominated by surface word to word association.

#### IV. REASSESSING SRN’S ABILITY TO GENERALIZE

The design of the experiment, both in [16] and in Simulation I aims to examine if SRNs can generalize in processing sentences that are made up of novel combinations of nouns and verbs. Taking a partial  $N=2$  testing set sentence  $n_{11}-v_{21}$  as an example. The combination of a group 1 noun and a group 2 verb in making up the sentence is novel to the network. For a SRN to exhibit the ability to generalize combinatorially, it has to acknowledge that:

- (1) nouns of different groups are in fact playing the same syntactic role with respect to verbs of different groups, and
- (2) verbs of different groups also share the same syntactic role with respect to nouns of different groups

From such knowledge the network should accept the testing set sentence as equally grammatical as a training set sentence, e.g.  $n_{11}-v_{11}$  or  $n_{21}-v_{21}$ , and makes the correct prediction that a noun should follow.

Before contrasting the ability of a language acquisition model with children on the ability to exhibit combinatorial productivity, we have to ask:

- Does Simulation I show SRNs' ability to achieve combinatorial productivity?
- How is that possible for SRNs to do so in the first place?
- And more critically, do the language and the environment that the networks are trained on provide conditions that are comparable to the situations in which children acquire language?

Results from Simulation I do not show full generalization as the GPE on testing set sentences are greater than the GPE on training set sentences. However, some degree of generalization is observed since SRNs' performance on testing set is significantly better than expected from a bi-gram model. We argue that this limited generalization observed in Simulation I is not due to the intrinsic incapability of SRN; rather it is due to the overly constrained exemplars of the language available to the network.

During the training process, there is no doubt that a SRN is able to develop syntactic knowledge of various groups of lexicon, as shown by the near zero GPE obtained in processing training set sentences. However, the construction of training set sentences systematically disallows the co-occurrence, within a sentence, of lexical items from different groups. Hence there is simply little basis for the network to develop a syntactic knowledge of general noun and general verb classes to satisfy (1) and (2), the only source of information available being the relative sentence position that all nouns and all verbs take.

## V. SIMULATION II

As we have argued that the limited generalization exhibited by SRNs is due to the setting of the training data but not the intrinsic incapability of the networks. Exemplars available to the networks in Simulation I do not provide a basis for generalization in the first place. To justify our argument, we carried out our second simulation where the training data are revised to establish a reasonable linguistic environment for the network.

### A. Training and Testing Data

The difference from the first experiment is that an extra group of training set sentences were incorporated<sup>1</sup>. Phase 1 to phase 3 training remained the same as in Simulation I, however during the last phase of training we added to the original training set an extra set of sentences which were generated from a *Group 5* lexicon. This extra set of *Group 5 training sentences* were generated from the Group 5 nouns ( $\mathbf{n}_5$ ) and the Group 5 verbs ( $\mathbf{v}_5$ ):

$$\mathbf{n}_5 = \{\mathbf{n}_{11}, \mathbf{n}_{21}, \mathbf{n}_{31}, \mathbf{n}_{41}\}$$

$$\mathbf{v}_5 = \{\mathbf{v}_{51}, \mathbf{v}_{52}\}$$

The  $\mathbf{v}_5$  verbs were new to the language, and were coded using the two reserved input and output neurons that we have

mentioned in section III. The Group 5 nouns,  $\mathbf{n}_5$ , however, contained the first noun from each of the original four groups. The construction of the training set sentences was the same as described in Simulation I and hence the set of Group 5 sentences are:

$$\begin{aligned} \text{Simple:} & \quad \{\mathbf{n}_{a1}-\mathbf{v}_{5x}-\mathbf{n}_{b1}-\#\}, \\ \text{Right-branching:} & \quad \{\mathbf{n}_{a1}-\mathbf{v}_{5x}-\mathbf{n}_{b1}-\text{who}-\mathbf{v}_{5y}-\mathbf{n}_{c1}-\#\}, \\ \text{Center-embedding:} & \quad \{\mathbf{n}_{a1}-\text{who}-\mathbf{n}_{b1}-\mathbf{v}_{5x}-\mathbf{v}_{5y}-\mathbf{n}_{c1}-\#\} \end{aligned}$$

where  $a,b,c = \{1..4\}$ ,  $x,y = \{1,2\}$

The Group 5 sentences were added as one fifth of the training set sentences which increased the total number of sentences fed to the networks to 80 000 during the last phase of training.

As for the testing sets, we have revised them such that trivial generalization due to the Group 5 sentences was avoided: sentences containing *any*  $\mathbf{n}_5$  nouns were removed from the original testing sets used in Simulation I. As a result, the revised testing sets were subsets of the original ones. Examples of  $N=3$  right-branching testing set sentences used in Simulation II were:

$$\begin{aligned} \mathbf{n}_{12}-\mathbf{v}_{32}-\mathbf{n}_{22}-\text{who}-\mathbf{v}_{11}-\mathbf{n}_{32}-\# \text{ and} \\ \mathbf{n}_{42}-\mathbf{v}_{32}-\mathbf{n}_{12}-\text{who}-\mathbf{v}_{42}-\mathbf{n}_{32}-\# \end{aligned}$$

### B. Results and Discussion of Simulation II

The results of training the networks this way are plotted in Fig. 5. Data points represent the mean GPE of 20 networks each tested with 100 testing set sentences.

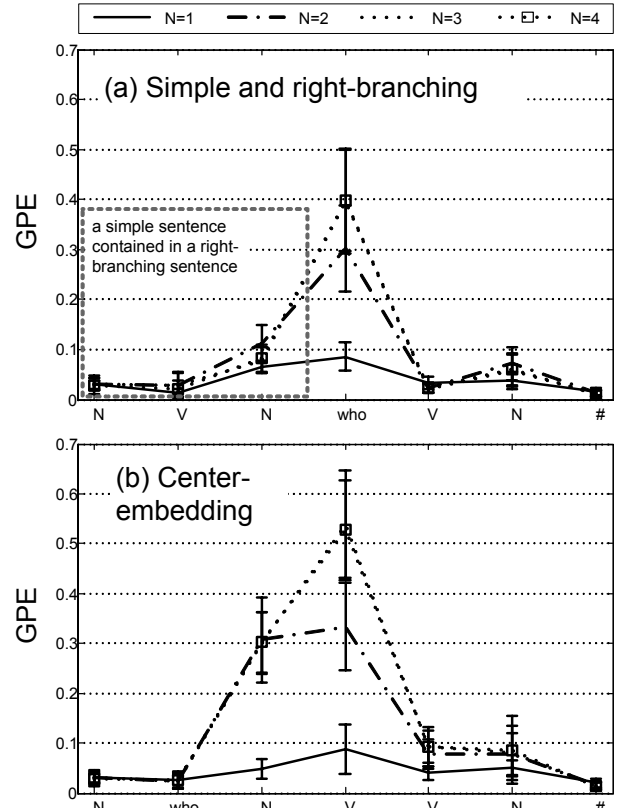


Fig. 5. Results of Sim II. Grammatical Prediction Error (GPE) in processing (a) simple, right-branching and (b) center-embedding sentences. Values are mean GPE of 20 networks each evaluated with 100 testing sentences.

By comparing the results of Simulations I and II (Fig. 2 and Fig. 5), an improvement on generalization is observed as the departure of the testing set GPE from the training set GPE is reduced significantly in most syntactic positions.

Analysis on the network’s raw output layer activation in processing a sentence reveals how improvement on generalization is possible. A snapshot of a SRN’s output layer activation pattern in processing an  $N=3$  testing set sentence is shown in Fig. 6. The GPE at this syntactic position, at the second noun of a simple or right-branching sentence, has been significantly reduced from 0.39 (Sim I, Fig. 3) to 0.05. This corresponds to the now-mature ability to process novel combinations of nouns and verbs by the networks trained in this revised simulation. We can also see that the incorrect prediction of verbs to follow is greatly reduced (to near-zero) indicating that the SRN is now much more sensitive to the underlying structural regularities than to the influence from immediately preceding word.

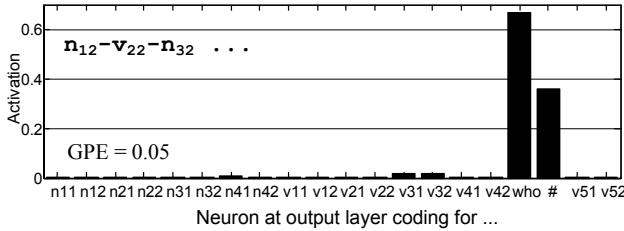


Fig. 6. Output layer activation pattern of, i.e. prediction made by, a SRN in processing an  $N=3$  right-branching sentence (cf Fig. 3).

For other syntactic positions where GPE on testing set sentences remains relatively high – at the position taken by the relative marker ‘who’ of right-branching sentences and the middle part of center-embedding sentences – there are due to the same reasons we have discussed in section III.E, namely the artifact of the artificial language where ‘who’ takes a verb-like position and the deviation from dominant N-V and V-N local word orders respectively.

### C. Rationale of Simulation II

We have to emphasize that bringing in the extra Group 5 sentences to train the networks does not weaken the evaluation of networks’ ability to generalize since training and testing sets remain to be non-overlapping. Rather, it provides realistic clues for the network to infer that there are some syntactic roles shared by nouns of different groups. Such inference is a result of the networks’ ability to induce underlying regularities of the language from exemplars.

The straight-forward way of mixing both the groups of nouns and the groups of verbs to form the set of extra sentences is avoided as it is crucial to clearly separate testing set sentences from training set sentences for the evaluation of SRNs’ ability to generalize. Moreover, GPE is measured at every sentence position, we make use of the set of novel  $v_5$  verbs to avoid the appearance of any part of a training set sentence in a testing ( $N>1$ ) set.

The design of the training and the testing data in Simulation II requires the networks, if there are indeed able to

generalize (attain low GPE on testing set sentences), to perform two level of generalization. With the exposure to Group 5 sentences, the SRNs have the chance to develop the knowledge that the  $n_5$  nouns share a similar syntactic role with respect to the novel  $v_5$  verbs. This knowledge has then to be transferred to the other verbs ( $v_1$  to  $v_4$ ), since GPE was not evaluated with sentences containing  $v_5$  verbs. Furthermore, the set of  $n_5$  nouns contains only half of the nouns from each group. Syntactic knowledge of the role played by the  $n_5$  nouns has also to be transferred to the other half, again since  $n_5$  nouns were not used during the evaluation of the network’s performance on testing set sentences.

Results of Simulation II confirm that the SRNs indeed exhibit the ability to induce syntactic knowledge from distributional information, to transfer such knowledge across isolated groups and to make generalization combinatorially.

## VI. DISCUSSION AND CONCLUSION

One of the merits of computational modeling, connectionist or non-connectionist, is in shedding light on linguistic theory. The simulation framework of van der Velde *et al.* [16] rightly addresses the issue of whether the SRN model exhibits human-like combinatorial productivity in acquiring a language. We consider the issue of combinatorial productivity to be another form of the learnability problem, as illustrated in Fig. 7.

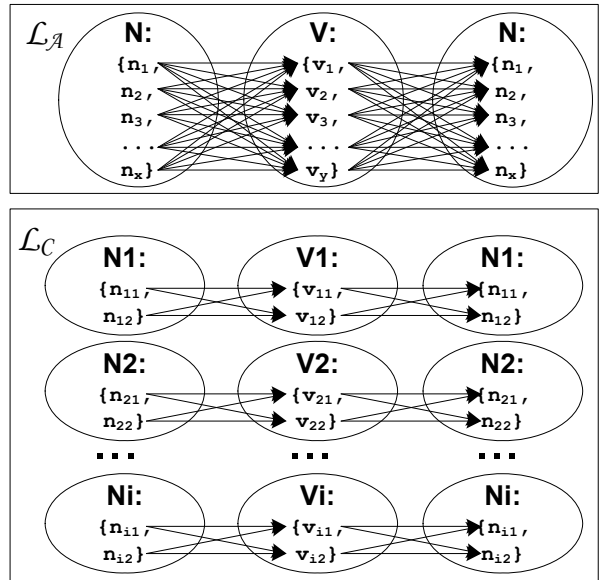


Fig. 7. Combinatorial productivity as a form of learnability problem: Can the model induce the grammaticality of sentences in  $L_A$  given that it has been only exposed to sentences of  $L_C$ ?  $L_A$  is the adult language learners ultimately arrived at.  $L_C$ , which is a constrained subset of  $L_A$ , is the set of exemplars available to the learners. An arrow indicates the construction of a grammatical sentence.

Suppose  $L_A$  is the target adult language that learners ultimately arrive at, for simplicity we consider only simple declarative sentences, where all possible combinations of nouns and verbs are grammatical. The language learners, however, are only exposed to a limited portion of them,  $L_C$ . It

has been argued that models based on statistical and/or inductive learning cannot generalize from  $L_C$  to  $L_A$ , in the other words  $L_A$  is not learnable from  $L_C$  unless rules or some rule-based innate biases have been incorporated into the learning algorithm [5, 6, 8].

In the context of this paper and in [16],  $L_C$  is the training set while  $L_A$  contains the testing sets for the evaluation of SRNs' ability to generalize combinatorially. Recall that GPE was used for such purpose. For a SRN to achieve low GPE on testing set sentences, it has to induce the grammaticality of the testing set sentences from the training set sentences (section IV). We argue, contrary to van der Velde, that SRNs can exhibit such productivity based on their ability to extract subtle distributional information available in the linguistic input.

We have reported results of two simulations regarding combinatorial productivity exhibited by SRNs. Our first simulation follows van der Velde's with respect to the construction of the training and the testing data. Results of the first simulation provide evidence arguing against the claim in [16] that SRNs lose track of testing set sentences, hence fail to generalize, and resort solely to bi-gram statistics.

We also disagree with their underestimation of the richness of information actually available to children that is not modeled in the training data. In a child language acquisition scenario, generalization is not expected to come out of the blue. It has to be based on numerous exposures [11, 22] to the target language. The intention of our second simulation is to provide a richer, more naturalistic linguistic environment as the basis for the networks to generalize while not neglecting the importance of giving the SRNs an objective evaluation of generalization. Results show that SRNs are indeed able to generalize combinatorially.

#### ACKNOWLEDGMENT

The authors would like to thank Thomas H. T. LEE, Tao GONG and Geng PANG for useful comments and suggestions. The research is supported by research grants from the RGC Hong Kong: CUHK-1224/02H and CUHK-1127/04H.

#### REFERENCES

- [1] D. E. Rumelhart and J. L. McClelland, *Parallel distributed processing: explorations in the microstructure of cognition*, vol. 1. Cambridge, Mass.: MIT Press, 1986.
- [2] J. L. McClelland and D. E. Rumelhart, *Parallel distributed processing: psychological and biological models*, vol. 2. Cambridge, Mass.: MIT Press, 1986.
- [3] J. Fodor and Z. Pylyshyn, "Connectionism and cognitive architecture: A critical analysis," in *Connections and symbols*, S. Pinker and J. Mehler, Eds. Cambridge, MA: MIT Press, 1988, pp. 3-71.
- [4] G. F. Marcus, "Connectionism: with or without rules? Response to J.L. McClelland and D.C. Plaut (1999)," *Trends in Cognitive Sciences*, vol. 3, pp. 168-170, 1999.
- [5] G. F. Marcus, "Poverty of the stimulus arguments," in *The MIT encyclopedia of the cognitive sciences*, R. A. Wilson and F. C. Keil, Eds. Cambridge, Mass.: MIT Press, 1999.

- [6] G. F. Marcus, S. Vijayan, S. Bandi Rao, and P. M. Vishton, "Rule Learning by Seven-Month-Old Infants," *Science*, vol. 283, pp. 77-80, 1999.
- [7] G. F. Marcus, "Can connectionism save constructivism," *Cognition*, vol. 66, pp. 153-182, 1998.
- [8] G. F. Marcus, *The algebraic mind: integrating connectionism and cognitive science*. Cambridge, Mass. ; London: MIT Press, 2001.
- [9] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179-211, 1990.
- [10] M. H. Christiansen and N. Chater, "Toward a connectionist model of recursion in human linguistic performance," *Cognitive Science*, vol. 23, pp. 157-205, 1999.
- [11] J. L. Elman, "Generalization, simple recurrent networks, and the emergence of structure," in *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, M. A. Gembacher and S. Derry, Eds. Mahway, NJ: Lawrence Erlbaum Associates, 1998.
- [12] J. L. Elman and E. Bates, "Acquiring language - Response," *Science*, vol. 276, pp. 1180-1180, 1997.
- [13] J. L. Elman, "Generalization, rules, and neural networks: a simulation of Marcus et al.," Retrieved 4th January, 2006, from <http://crl.ucsd.edu/~elman/Papers/MVRVsimulation.html>, 1999.
- [14] M. S. Seidenberg, J. L. Elman, M. Negishi, P. D. Eimas, and G. F. Marcus, "Do Infants Learn Grammar with Algebra or Statistics?," *Science*, vol. 284, pp. 433, 1999.
- [15] J. L. McClelland and D. C. Plaut, "Does generalization in infant learning implicate abstract algebra-like rules?," *Trends in Cognitive Sciences*, vol. 3, pp. 166-168, 1999.
- [16] F. van der Velde, G. T. van der Voort van der Kleij, and M. de Kamps, "Lack of combinatorial productivity in language processing with simple recurrent networks," *Connection Science*, vol. 16, pp. 21-46, 2004.
- [17] F. van der Velde, "Modelling language development and evolution with the benefit of hindsight," *Connection Science*, vol. 17, pp. 361-379, 2005.
- [18] S. Pinker, *Language learnability and language development*, 2nd ed. Cambridge, Mass.: Harvard University Press, 1996.
- [19] J. L. Elman, *Rethinking innateness: a connectionist perspective on development*. Cambridge, Mass.: MIT Press, 1996.
- [20] J. L. Elman, "Learning and development in neural networks: the importance of starting small," *Cognition*, vol. 48, pp. 71-99, 1993.
- [21] M. H. Christiansen and J. T. Devlin, "Recursive inconsistencies are hard to learn: A connectionist perspective on universal word order correlations," in *Proceedings of the 19th Annual Cognitive Science Society Conference*. Mahwah, NJ: Lawrence Erlbaum, 1997, pp. 113-118.
- [22] M. Tomasello, "The item-based nature of children's early syntactic development," in *Language development: the essential readings*, M. Tomasello and E. Bates, Eds. Malden, Mass.: Blackwell Publishers, 2001, pp. 169-186.

---

<sup>1</sup> Though in one of the simulations reported in van der Velde *et al.* training set sentences composed of lexical items from mixed groups had been incorporated, critical details of the setting are not available. Moreover, training and testing set sentences were not as disentangled as they should be, making their results less informative.