Recognition and Generation of Sentences through Self-organizing Linguistic Hierarchy Using MTRNN

Wataru Hinoshita[1], Hiroaki Arie[2], Jun Tani[2],
Tetsuya Ogata[1], and Hiroshi G. Okuno[1]

[1] Graduate School of Informatics, Kyoto Univ.
Engineering Building #10, Sakyo, Kyoto, 606-8501, Japan
[2] Brain Science Institute, RIKEN
2-1 Hirosawa, Wako-shi, Saitama, 351-0198, Japan

Abstract. We show that a Multiple Timescale Recurrent Neural Network (MTRNN) can acquire the capabilities of recognizing and generating sentences by self-organizing a hierarchical linguistic structure. There have been many studies aimed at finding whether a neural system such as the brain can acquire languages without innate linguistic faculties. These studies have found that some kinds of recurrent neural networks could learn grammar. However, these models could not acquire the capability of deterministically generating various sentences, which is an essential part of language functions. In addition, the existing models require a word set in advance to learn the grammar. Learning languages without previous knowledge about words requires the capability of hierarchical composition such as characters to words and words to sentences, which is the essence of the rich expressiveness of languages. In our experiment, we trained our model to learn language using only a sentence set without any previous knowledge about words or grammar. Our experimental results demonstrated that the model could acquire the capabilities of recognizing and deterministically generating grammatical sentences even if they were not learned. The analysis of neural activations in our model revealed that the MTRNN had self-organized the linguistic structure hierarchically by taking advantage of differences in the time scale among its neurons, more concretely, neurons that change the fastest represented "characters," those that change more slowly represented "words," and those that change the slowest represented "sentences."

1 Introduction

The question of whether a neural system such as the brain can acquire a creative command of languages without innate linguistic capabilities has been the object of discussion for many years. Chomsky [1] claimed that there should be an innate faculty for language in the human brain because of the "poverty of the stimulus" argument. This argument is that the linguistic stimuli that a child can experience in reality are not enough in either quantity or quality for him or her to

induce general rules of the language from these. Linguists who support nativism emphasize the fact that children can learn to recognize and generate diverse new grammatical sentences using only limited linguistic stimuli, which include virtually no evidence of what is ungrammatical. However, the recent progress made in analyzing dynamical systems and chaos [2] has revealed that diverse complex patterns can emerge from a few input patterns. Thus, the controversy between nativists and experientialists about language acquisition is not over.

Many studies have aimed at revealing whether neural systems can acquire languages using neural network models [2,3,4,5,6,7,8]. Pollac [2] showed the phase transition of non-linear dynamical systems can lead to generative capacity of language using his higher-order-recurrent neural network, but his model required both positive and negative examples of language to learn the rules. Elman [3,4,5] proposed the Simple Recurrent Network (SRN) and showed that it could self-organize grammar using only a sentence set. However, this model could not deterministically generate sentences, but could predict the possibilities of the next word from those that had been input up to that step. Sugita and Tani [9] and Ogata et al. [10] used an RNN model with Parametric Bias (RNNPB) [11] for language learning. These models could learn multiple sequences and deterministically generate them by changing the parametric bias. However, they dealt with simple sentences composed of two or three words, because the models had difficulty learning long complex sequences. Thus, the question as to whether a neural system can acquire generative capacity from a sentence set still remains unanswered. This question is crucial to the problem of language acquisition because generative capacity is an essential part of human language functions.

Existing RNN models for language acquisition such as SRN and RNNPB require a predetermined word set to learn the grammar [3,4,5,6]. Learning languages without such previous knowledge requires the capability to hierarchically compose characters into words, and words into sentences. This capability is essential for dealing with the diversity of expressions in language. Thus, it is also important to find whether a neural system can acquire such hierarchical structures.

We discovered that a Multiple Timescale Recurrent Neural Network (MTRNN) [12] can acquire the capabilities of recognizing and generating sentences even if they are not learned through the self-organization of the linguistic hierarchical structure. We trained an MTRNN using only a sentence set without any previous knowledge about the lexicon or grammar.

2   Language Learning Model

Our language learning model is based on an MTRNN, an extended RNN model proposed by Yamashita and Tani [12]. An MTRNN deals with sequences by calculating the next state $S(t+1)$ from the current state $S(t)$ and the contextual information stored in their neurons. The model is composed of several neuron groups, each with an associated time constant. If the neurons have a larger time constant, their states change more slowly. The time scale difference causes the information to be hierarchically coded. An MTRNN can deterministically generate sequences depending on the initial states of certain context nodes. Moreover,

given a sequence, the model can calculate the initial states from which it generates the target sequence. Therefore, this model can be used as the recognizer and generator of the sequences. The initial state space is self-organized based on the dynamical structure among the training sequences. Thus, the model deals with even unknown sequences by generalizing the training sequences.

Figure 1 shows an overview of our language learning model that has three neuron groups, which are input-output (IO), Fast Context (Cf), and Slow Context (Cs) groups, in increasing order of time constant (τ). The IO has 30 nodes and each of them corresponds to one of the characters from the 26 letters in the alphabet ('a' to 'z') and four other symbols (space, period, comma, and question mark). Cf has 40 nodes and Cs has 11. We choose six neurons from Cs to be used as the Controlling Slow Context (Csc), whose initial states determine the sequence. In our model, a sentence is represented as a sequence of IO activations corresponding to the characters. The model learns to predict the next IO activation from the activations up to that point. Therefore, we only need to use a set of sentences to train our model. Figure 2 shows an example of the training sequence for this model.

The activation value of the i-th neuron at step t ($y_{t,i}$) is calculated as follows.

$$
y_{t,i} = \begin{cases} \dfrac{\exp(u_{t,i} + b_i)}{\displaystyle\sum_{j \in I_{IO}} \exp(u_{t,j} + b_j)} & \cdots (i \in I_{IO}) \\[4ex] \dfrac{1}{1 + \exp(-(u_{t,i} + b_i))} & \cdots (i \notin I_{IO}) \end{cases} \tag{1}
$$

$$
u_{t,i} = \begin{cases} 0 & \cdots (t = 0 \wedge i \notin I_{Csc}) \\ Csc_{0,i} & \cdots (t = 0 \wedge i \in I_{Csc}) \\ \left(1 - \dfrac{1}{\tau_i}\right)u_{t-1,i} + \dfrac{1}{\tau_i}\left[\displaystyle\sum_{j \in I_{all}} w_{ij}x_{t,j}\right] & \cdots \text{(otherwise)} \end{cases} \tag{2}
$$

$$
x_{t,j} = y_{t-1,j} \qquad\qquad \cdots (t \geq 1) \tag{3}
$$

$I_{IO}, I_{Cf}, I_{Cs}, I_{Csc}$: neuron index set of each group ($I_{Csc} \subset I_{Cs}$)

I_{all} : $I_{IO} \cup I_{Cf} \cup I_{Cs}$

$u_{t,i}$: internal state of i-th neuron at step t

b_i : bias of i-th neuron

$Csc_{0,i}$: initial state that controls MTRNN

τ_i : time constant of i-th neuron

w_{ij} : connection weight from j-th neuron to i-th neuron

$w_{ij} = 0 \cdots (i \in I_{IO} \wedge j \in I_{Cs}) \vee (i \in I_{Cs} \wedge j \in I_{IO})$

$x_{j,t}$: input from j-th neuron at step t

The connection weights (w_{ij}), biases (b_i), and initial states ($Ccs_{0,i}$) are updated using the Back Propagation Through Time (BPTT) algorithm [13] as follows.

$$w_{ij}^{(n+1)} = w_{ij}^{(n)} \quad -\eta\frac{\partial E}{\partial w_{ij}} \quad = w_{ij}^{(n)} \quad -\frac{\eta}{\tau_i}\sum_t x_{t,j}\frac{\partial E}{\partial u_{t,i}} \tag{4}$$

$$b_i^{(n+1)} = b_i^{(n)} \quad -\beta\frac{\partial E}{\partial b_i} \quad = b_i^{(n)} \quad -\beta\sum_t \frac{\partial E}{\partial u_{t,i}} \tag{5}$$

$$Ccs_{0,i}^{(n+1)} = Ccs_{0,i}^{(n)} -\alpha\frac{\partial E}{\partial Ccs_{0,i}} = Ccs_{0,i}^{(n)} -\alpha\frac{\partial E}{\partial u_{0,i}} \qquad \cdots (i \in I_{Csc}) \tag{6}$$

$$E = \sum_t \sum_{i \in I_{IO}} y_{t,i}^* \cdot \log\left(\frac{y_{t,i}^*}{y_{t,i}}\right) \tag{7}$$

$$\frac{\partial E}{\partial u_{t,i}} = \begin{cases} y_{t,i} - y_{t,i}^* + (1 - \frac{1}{\tau_i})\frac{\partial E}{\partial u_{t+1,i}} & \cdots (i \in I_{IO}) \\ y_{t,i}(1 - y_{t,i})\sum_{k \in I_{all}} \frac{w_{ki}}{\tau_k}\frac{\partial E}{\partial u_{t+1,k}} + (1 - \frac{1}{\tau_i})\frac{\partial E}{\partial u_{t+1,i}} & \cdots (otherwise) \end{cases} \tag{8}$$

n : number of iterations in updating process

E : prediction error

$y_{t,i}^*$: value of current training sequence for i-th neuron at step t

η, β, α : learning rate constant

When using the BPTT algorithm, the input values $(x_{t,j})$ of IO are calculated along with the feedback from the training sequence using the following equation instead of (3).

$$x_{t,j} = (1 - r) \times y_{t-1,j} + r \times y_{t-1,j}^* \qquad \cdots (t \geq 1 \wedge j \in I_{IO}) \tag{9}$$
$$r \ : \ \text{feedback rate } (0 \leq r \leq 1)$$

The initial Csc states determine the MTRNN's behavior. Thus, we define a set of initial states (Csc_0) as follows.

$$Csc_0 = \{(i, Ccs_{0,i}) | i \in I_{Csc}\} \tag{10}$$

Csc_0 is independently prepared for each training sequence while the network weights (connection weights and biases) are shared by all the sequences. The initial state space is self-organized based on the dynamical structure among the training sequences through a process where the network weights and Csc_0 are simultaneously updated.

To recognize a sequence, the Csc_0 representing the target sequence is calculated using the BPTT with fixed network weights from (6). In this recognition phase, the input values of IO are calculated by using (9) if the value of the target sequence is given, otherwise they are calculated by using (3). Thus, the MTRNN can recognize sequences even if only partial information is given.

A sequence is generated by recursively executing a forward calculation ((1), (2), and (3)) using a Csc_0 that represents the target sequence.
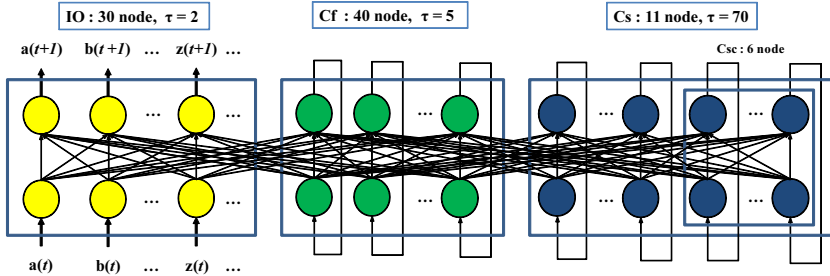
Fig. 1. Overview of Language Learning MTRNN: $a(t)$ is activation value of neuron corresponding to 'a'. The others $(b(t), ..., z(t), ...)$ are defined in the same way. The sentences are represented by successive activations of IO neurons.
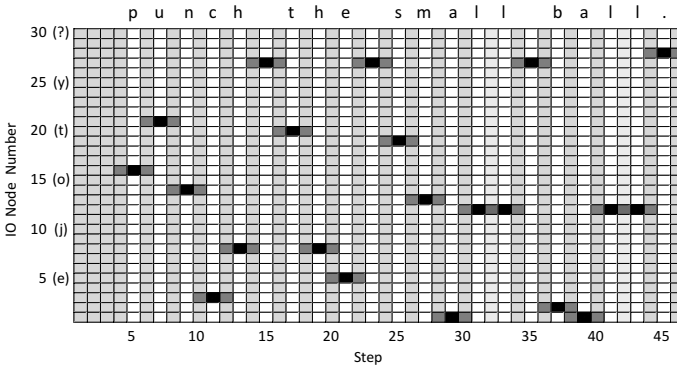

Fig. 2. Example of training sequence: "punch the small ball"

3   Language Learning Experiment

We trained the MTRNN to learn language using only a sentence set, without any previous knowledge about the words or grammar, but only the character set with each character corresponding to one of the IO neurons. This experiment was aimed at finding whether the MTRNN could learn to recognize and generate sentences even if they were not included in the training sentences. If the model could acquire the necessary capabilities, the linguistic structure would have been self-organized by MTRNN from the sentence set.

In this experiment, we used a very small language set to make it possible to analyze the linguistic structure self-organized in the MTRNN. Our language set contained 17 words in seven categories (Table 1) and a regular grammar that consisted of nine rules (Table 2). (It was designed for robot tasks.)

3.1   Experimental Procedure

1. Derive 100 different sentences from the regular grammar.
2. Train the MTRNN using the first 80 sentences.

Table 1. Lexicon   **Table 2.** Grammar

Category	Nonterminal symbol	Words
Verb (intransitive)	V_I	jump, run, walk
Verb (transitive)	V_T	kick, punch, touch
Noun	N	ball, box
Article	ART	a, the
Adverb	ADV	quickly, slowly
Adjective (size)	ADJ_S	big, small
Adjective (color)	ADJ_C	blue, red, yellow

S → V_I
S → V_I ADV
S → V_T NP
S → V_T NP ADV

NP → ART N
NP → ART ADJ N

ADJ → ADJ_S
ADJ → ADJ_C
ADJ → ADJ_S ADJ_C

3. Test the trained MTRNN's capabilities using both the 80 sentences and the remaining 20 sentences. The testing procedure was involved three steps.
   (i) Recognition: Calculate Csc_0 from a sentence.
   (ii) Generation : Generate a sentence from the Ccs_0 gained in (i).
   (iii) Comparison: Compare the original and generated sentence.
4. Test the MTRNN using another 20 sentences that are ungrammatical as a control experiment.

The calculation of Csc_0 by the BPTT from (6) sometimes falls to a local minimum in the recognition phase. Therefore, we calculate it 20 times while changing the initial value in the updating process $(Ccs_{0,i}^{(0)})$, and choose the result with the lowest error E (cf. (7)).

3.2   Results

We found that our model could correctly generate 98 of 100 grammatical sentences. To correctly generate a sentence, a stable trajectory representing the sentence should be formed in the dynamical system of the MTRNN and its Csc_0 should be properly embedded into the initial state space. We have listed the sentences that the model failed to generate in Table 3.

We also found that the generated sentences did not match the originals for all of the 20 ungrammatical sentences in the control experiment. This is because the recognition error (E (cf. (7)) in the recognition phase) did not adequately decrease. Indeed, the average recognition error for the 20 ungrammatical sentences was about 22 times that of the 20 unknown grammatical sentences.

These results revealed that our model self-organized the linguistic structure using only the sentence set.

Table 3. Failed sentences

Sentence number	Original sentence	Generated sentence
082 (not learned)	"kick a big yellow box."	"kick a sillylllow box."
100 (not learned)	"jump quickly."	"jump slowloxl"

4   Analysis

We claim that our model hierarchically self-organized a linguistic structure, more precisely that IO neuron activation represents the "characters," Cf represents the "words," and Cs represents the "sentences." We illustrate the basis of this argument in this section by analyzing our model.

We analyzed the neural activation patterns when the MTRNN generated sentences to reveal the linguistic structures self-organized in the MTRNN. We used principle component analysis (PCA) in our analysis. We have given some examples of the transitions of Cf neural activation in Fig. 3, and those of Cs in Fig. 4. The three activation patterns in these figures correspond to the sentences, "walk slowly.," "punch the yellow box slowly.," and "kick a small yellow ball.." We have summarized the results of the analysis for each neuron group below.

IO : Each IO neuron corresponds to a character. Thus, their activation patterns obviously represent the sequences of the "characters."
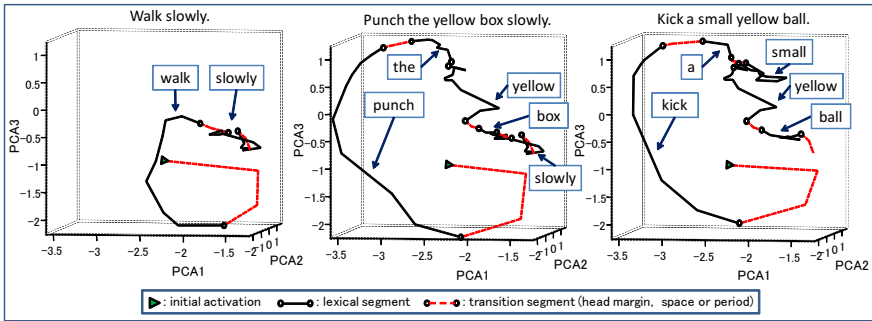

Fig. 3. Transitions of Cf activation : dimensions are reduced from 40 to 3 by PCA (the total contribution rate is 86%). The same words are represented as the same trajectories, and the words in the same categories are represented in similar ways.
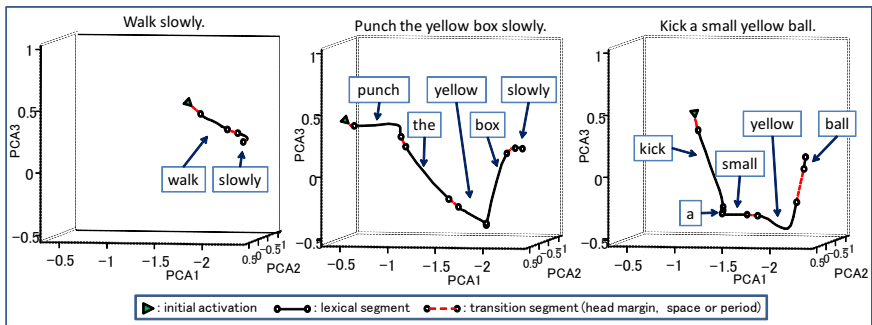

Fig. 4. Transitions of Cs activation : dimensions are reduced from 11 to 3 by PCA (the total contribution rate is 95%). In different sentences, even the same words are represented in different ways.
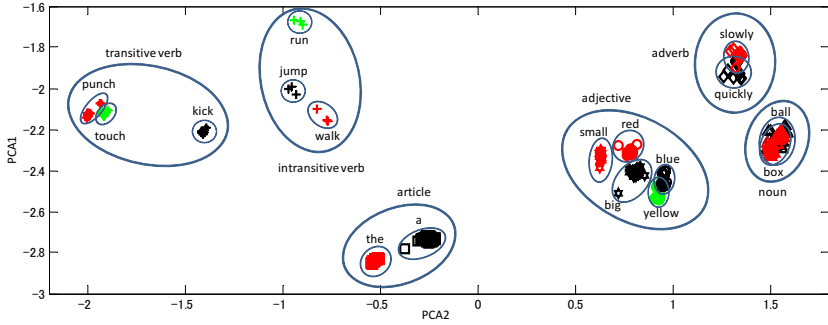
Fig. 5. Cf activation in first step of each word: words are clustered based on their categories
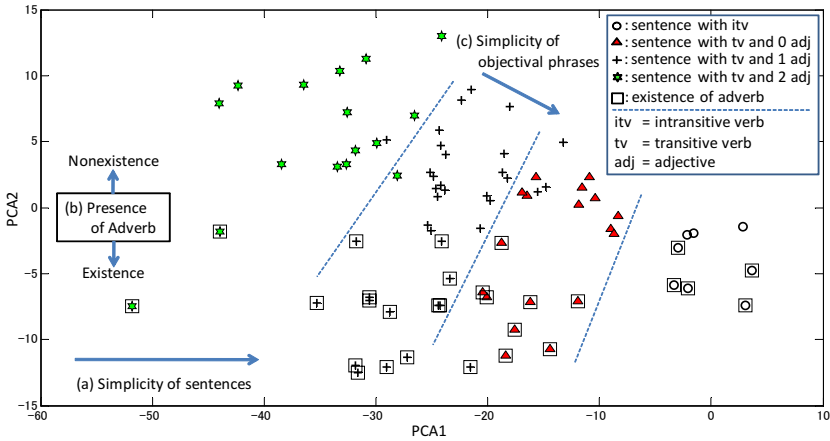

Fig. 6. Initial state of Cs (Csc_0): dimensions are reduced from 6 to 2 by PCA (total contribution rate is 90%). The sentences are clustered based on their grammatical structure. (a) The value of PCA1 seems to be negatively correlated with the number of words in sentences, i.e., the complexity of sentences. (b) There seems to be a PCA2 threshold that separates whether a sentence has an adverb or not. (c) Focusing on the number of words in an objective phrase, there seems to be an axis correlated with it.

Cf : We claim that Cf activation represents the "words" including their grammatical information. Our claim is based on the following facts, which are found in Fig. 3.

1. The correspondence between characters and activations disappeared. This can easily be confirmed since the activation patterns are different even if the characters are the same.

2. The same words are represented by the same trajectories (e.g., "yellow" in the center and to the right of the figure).

3. The words in the same category are represented in a similar way (e.g., "punch" in the center of the figure and "kick" to the right of the figure).

4. The first and the last steps of the words are clustered by their grammatical roles, and the grammatical associativity between categories is represented by their closeness (e.g., an intransitive verb ("walk") ends near the start of adverbs, but transitive verbs ("punch" and "kick") end near the start of articles).

We also have shown the Cf activations in the first step of each word in all the sentences in Fig. 5. This clearly illustrates that words are clustered by their grammatical roles.

Cs : We claim that the Cs activation represents the "sentences." These are two main bases for our claim.

1. The correspondence between words and activations disappeared. Even the same words in different sentences are represented in different ways in Fig. 4 (e.g., "yellow" in the center and to the right of the figure).
2. The initial states of Cs (Csc_0) are clustered mainly by the grammatical structure of the sentences (Fig. 6). The grammatical structure is featured by both the existence of an adverb and the complexity of the objectival phrase. The complexity of the objectival phrase increases in the following order.
   (i) sentence with a intransitive verb (e.g., "walk.")
   (ii) sentence with a transitive verb and no adjectives (e.g., "kick *a box.*")
   (iii) sentence with a transitive verb and an adjective
       (e.g., "kick *a red box.*")
   (iv) sentence with a transitive verb and two adjectives
       (e.g., "kick *a big red box.*")

5   Conclusion

We reported on language learning achieved by using an MTRNN. We trained the model to learn language using only a sentence set without any previous knowledge about the words or grammar, but only about the character set. As a result of our experiment, we found that the model could acquire capabilities of recognizing and generating sentences even if they were not learned. Therefore, we found that our model could self-organize linguistic structures by generalizing a sentence set. To discover this structure, we analyzed the neural activation patterns in each neuron group. As a result of the analysis, we found that our model hierarchically self-organized language taking advantage of the difference in time scales among neuron groups. More precisely, the IO neurons represented the "characters," the Cf neurons represented the "words," and the Cs neurons represented the "sentences." The alternative view was that the network weights of IO coded the sequence of characters for each word, and those of the Cf coded the grammars as the associativity between words, and those of the Cs coded the separate sentences themselves. The model recognizes and generates sentences through the interaction between these three levels.

We proved in an experiment that a neural system such as a MTRNN can self-organize the hierarchical structure of language (e.g., characters → words → sentences) by generalizing a sentence set, and it can recognize and generate new

sentences using the structure. This implies that the requirements for language acquisition are not innate faculties of a language, but appropriate architectures of a neural system (e.g., differences in the time scale). Of course, this is not direct evidence for experientialism in language acquisition, but important knowledge supporting that theory.

In future work, we intend to deal with language acquisition from the viewpoint of the interaction between linguistic cognition and other types of cognition (this viewpoint is that of cognitive linguists). Specifically, we are going to connect the language MTRNN with another MTRNN for the sensori-motor flow of a robot. We expect the robot to acquire language grounded on its sensori-motor cognition using the dynamical interaction between the two MTRNNs.

References

1. Chomsky, N.: Barrier. MIT Press, Cambridge (1986)
2. Pollack, J.B.: The induction of dynamical recognizers. Machine Learning 7(2-3), 227–252 (1991)
3. Elman, J.L.: Finding structure in time. Cognitive Science 14, 179–211 (1990)
4. Elman, J.L.: Distributed representations, simple recurrent networks, and grammatical structure. Machine Learning 7(2-3), 195–225 (1991)
5. Elman, J.L.: Language as a dynamical system. In: Port, R., van Gelder, T. (eds.) Mind as Motion: Explorations in the Dynamics of Cognition, pp. 195–223. MIT Press, Cambridge (1995)
6. Weckerly, J., Elman, J.L.: A pdp approach to processing center-embedded sentences. In: Fourteenth Annual Conference of the Cognitive Science Society, vol. 14, pp. 414–419. Routledge, New York (1992)
7. Cleeremans, A., Servan-Schreiber, D., McClelland, J.L.: Finite state automata and simple recurrent networks. Neural Computation 1(3), 372–381 (1989)
8. Giles, C.L., Miller, C.B., Chen, D., Chen, H.H., Sun, G.Z., Lee, Y.C.: Learning and extracting finite state automata with second-order recurrent neural networks. Neural Computation 4(3), 393–405 (1992)
9. Sugita, Y., Tani, J.: Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. Adaptive Behavior 13(1), 33–52 (2005)
10. Ogata, T., Murase, M., Tani, J., Komatani, K., Okuno, H.G.: Two-way translation of compound sentences and arm motions by recurrent neural networks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), pp. 1858–1863 (2007)
11. Tani, J., Ito, M.: Self-organization of behavioral primitives as multiple attractor dynamics: A robot experiment. IEEE Trans. on Systems, Man, and Cybernetics Part A: Systems and Humans 33(4), 481–488 (2003)
12. Yamashita, Y., Tani, J.: Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment. PLoS Comput. Biol. 4 (2008)
13. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: 8. In: Learning internal representations by error propagation, pp. 318–362. MIT Press, Cambridge (1986)