

THE CONDITION OF RECURSIVE COMBINATION IN THE EVOLUTION OF REINFORCEMENT LEARNING AGENTS

GENTA TOYA^{*1}, RIE ASANO², and TAKASHI HASHIMOTO³

^{*}Corresponding Author: toyagent@protonmail.com

¹Graduate School of Arts and Sciences, The University of Tokyo, Meguro, Japan

²Department of Systematic Musicology, University of Cologne, Cologne, Germany

³School of Knowledge Science, Japan Advanced Institute of Science and Technology, Nomi, Japan

Language consists of multiple components with different evolutionary origins (Boeckx, 2013). Beside its central role in communication based on the capacity for intention sharing (Tomasello, 1999), another important component of language is its hierarchical structure. Human language requires not only a sequential operation but also a recursive operation generating hierarchical expressions by combining two elements, e.g., X and Y, into a unit $\{X, Y\}$ which is again combined into another unit $\{Z, \{X, Y\}\}$ (Chomsky, 1995). How and why did the recursive combination (hereafter RC) evolve? RC is significantly observed in object manipulation by humans (Greenfield, 1991) as well as by captive chimpanzees (Matsuzawa, 1991; Hayashi, 2007). Because object manipulation as a necessary skill for tool-use and -making can increase the fitness of an individual without cooperating with others, it is hypothesized that the RC of objects or actions evolved as a precursor of RC of lexical items or symbols (Fujita, 2017). We pursue this hypothesis of the evolutionary scenario that RC in motor control extended to RC in language in the course of human evolution.

In our simulation study, we model learning organisms as reinforcement learning agents (Sutton & Barto, 2018). By using agent-based modeling and evolutionary simulation, we investigate how and why RC evolved in learning organisms. In this simulation, agents, which are equipped with a reinforcement learning algorithm, explore and learn the process of tool making. Tool-making is implemented as a combination of elements through state transitions based on Q-values. In addition, the hyperparameters (learning rate α , exploration rate ϵ , and time discount rate γ) of the neural network encoding the reinforcement learning

and Q-values are explored by the genetic algorithm. The way agents make tools can be classified into two categories. One is the non-RC type production, in which elements are sequentially combined into a single object. The other is RC type production, where the combined object is re-combined with another object using a stack that can be acquired evolutionarily. The varieties of products that can be made with either production method remain the same. We set the reward function of the reinforcement learning algorithm so that the agents are rewarded more for making novel products. This corresponds to the phenomena that the invention of tools allows access to new resources (Arthur, 2009). We define the fitness function of the genetic algorithm so that the total rewards of an agent in a generation is discounted by the depth of the stack as a cost. In other words, this is a more favorable setting for non-RC than for RC because RC is more costly due to the additional use of the stack.

We found out that critical parameters for the emergence of RC include the cost of the stack, the reward discount rate, element types, and product length. If the cost of the stack, which is necessary for RC, is low, RC emerges because using RC in addition to the reinforcement learning is more advantageous than using reinforcement learning only. If the reward discount rate of producing the same product is low, it is less adaptive to produce as diverse products as possible. In this case, RC emerges because reinforcement learning with a high exploration rate ϵ is a less valuable strategy (Figure 1). If the types of elements are more and the product length is short, RC emerges because it is more effective than reinforcement learning with a low time discount rate γ .

In sum, in solving the exploration and exploitation problem, RC emerges in an environment where exploitation and exploration should be balanced. Moreover, the environment, in which attending to multiple states (i.e., to both workspace and stack) is not costly, facilitates the emergence of RC.

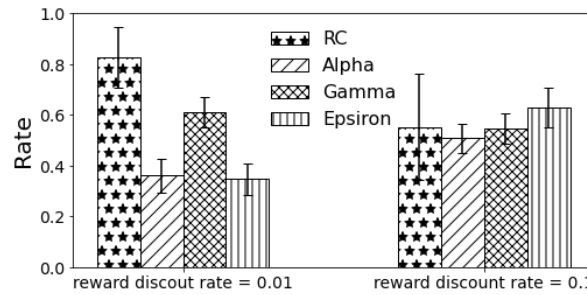


Figure 1. Reward discount rate dependency of the population share of RC users, hyperparameter of Q-learning (Alpha_QL, Gamma_QL, Epsilon_QL) at the 1,000th generation (average of 50 trials).

References

- Arthur, B. W. (2009). *The nature of technology: What it is and how it evolves*. The Free Press, NY: Simon & Schuster.
- Boeckx, C. (2013). Merge: Biolinguistic considerations. *English Linguistics* 30, 463-484.
- Chomsky, N. (1995). Bare Phrase Structure. In H. Campos. & P. Kempchinsky (Eds.), *Evolution and Revolution in Linguistic Theory* (pp. 1-15). Washington DC: Georgetown University Press.
- Fujita, K. (2017). On the parallel evolution of syntax and lexicon: A Merge-only view. *The Journal of Neurolinguistics*, 43, 178–192.
- Greenfield, P. (1991). Language, tools and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior. *Behavioral and Brain Science*, 14, 531–595.
- Hayashi, M. (2007). A New notation system of object manipulation in the nesting-cup task for chimpanzees and humans, *Cortex*, 43, 308–318.
- Matsuzawa, T. (1991). Nesting cups and metatools in chimpanzees. *Behavioral and Brain Science*, 14: 570–571
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT press.
- Tomasello, M. (1999). *The cultural origins of human cognition*. Cambridge, MA: Harvard University Press.