Evolution of Symbolic Grammar Systems

Takashi Hashimoto * and Takashi Ikegami **

Institute of Physics, College of Arts and Sciences, University of Tokyo,
Komaba 3-8-1,Meguro-ku, Tokyo 153, Japan

Abstract. Evolution of symbolic language and grammar is studied in a network model. Language is expressed by words, i.e. strings of symbols, which are generated by agents with their own symbolic grammar system. By deriving and accepting words, the agents communicate with each other. An agent which can derive less frequent and less acceptable words and accept words in less computational time will have higher scores. Grammars of agents can evolve by mutationally processes, where higher scored agents have more chances to breed their offsprings with improved grammar system. Complexity and diversity of words increase in time. It is found that the module type evolution and the emergence of loop structure enhance the evolution. Furthermore, ensemble structure (net-grammar) emerges from interaction among individual grammar systems. A net-grammar restricts structures of individual grammar and determines their evolutionary pathway.

1 Introduction

Linguistic expressions are quite complex but may not be random. It is commonly assumed that one has to have an internal knowledge (hereafter *individual grammar*) of one's language when one can derive and recognize appropriately structured expressions. On the other hand, linguistic expressions are determined and restricted by a community of agents. Language is used by many speakers, not just a single speaker, the language as a whole is produced through interaction among various individual grammars. In this respect, a network determining grammar may be more important for linguistic expressions than internal knowledge. An individual grammar does not have a static form but dynamically changes: it can undergo changes induced by interactions with physical and cultural environment or conversations with other people. We have to discuss how is the grammar of a language is constructed through interaction among individual grammars and how does diversity and complexity of individual grammar evolve?

In the present paper, we will study an evolution of grammar in network through an agent model, where each agent has its own grammar and it communicates with. In our model the individual grammar is expressed by a symbolic generative grammar. When each grammar changes, the set of words it permits

* e-mail: toshiwo@sacral.c.u-tokyo.ac.jp
** e-mail: ikeg@sacral.c.u-tokyo.ac.jp

can change. The evolution of diversity of spoken words and such generative grammars will be discussed. Adequate automaton can accept the set of words which is accepted by a given symbolic grammar [?]. Hence the diversity of spoken words of a symbolic grammar system can be studied in terms of computational ability of automata.

According to N. Chomsky, the corresponding computational ability of symbolic language system is categorized into four different classes with respect to its grammar structure as follows [?]:

type 0 phrase structure grammar
type 1 context sensitive grammar
type 2 context free grammar
type 3 regular grammar.

A grammar in an upper hierarchy class generates a larger set of words than ones lower in the hierarchy. For example, a word set $\{0^n1^n | n \geq 1\}$ can not be derived by regular grammar (type 3) but can be derived by context free grammar (type 2) or ones even higher in the hierarchy, where xy is a concatenation of symbol x and y and x^n is n times concatenation of symbol x.

For practical situations, we have to deal with finite length of words with finite deriving processes. If we deal only with the finite set of symbols, i.e. $\{0^n1^n | N \geq n \geq 1\}$, this hierarchy does not always work. In computation theory, computational time to derive words are not bounded and no ensemble effect is considered. In this paper, we study practical ability to speak and recognize words. We need to figure out what kind of grammar has the practical ability to derive and accept words in finite time. Namely, the computational ability of a symbolic grammar and hierarchy will be studied in an ensemble of communicating agents.

Relations between different levels can only be clarified in a network and evolutionary context. If an upper structure in a network evolves to constrain individual grammars, we call it a *net-grammar*. A net-grammar system emerges from interactions between individual grammar systems rather than from one grammar system. A relationship between a net-grammar system and individual grammar systems will be discussed in this paper.

By taking each individual grammar as genotype and the set of generated words as the phenotype, we can regard a symbolic grammar system as a genetic system. In this paper, individual grammars can evolve through mutationally processes as well as genetic evolution.

Furthermore, autonomy of language is our main concern. It has been assumed that complexity of language is a mere reflection of complexity in the world we live, just as complexity of living systems is said to be the reflection of their complex environments. MacLennan has studied the communication among agents with simple rules[?]. His agents live in a particular local environment and communicate with each other by emitting signals. Those signals correspond to their external objects. Werner and Dyer have discussed the evolution of communication among the spatially distributed agents [?]. However, we believe that

even without locality in space/information, systems can evolve and diversify their phenotype and genotype by some internal mechanisms. Examples can be found in evolutionary game [?, ?], Tierra world [?, ?]. We study evolution and diversification of sentences and grammars without external environments. Only conversation among agents can evolve grammar structure.

2 Modeling

2.1 Communication between Symbolic Grammar Systems

Agent. We express a communicating agent with a *generative grammar* by an ordered four-tuple:

$$G_i = (V_N, V_T, F_i, S) \ . \tag{1}$$

All agent have the same sets of nonterminal and terminal symbols, that is $V_N = \{S, A, B\}$, $V_T = \{0, 1\}$ respectively. Each agent is identified by index i. A symbol F_i is a set of rewriting rules peculiar to each agent, which is a finite set of ordered pair (α, β). The elements (α, β) in F are called *rewriting rules* and will be written $\alpha \to \beta$. Here, α is a symbol over V_N. And β is an arbitrary finite string of symbols over $V_N \cup V_T$ not including the same symbol as α. The type of grammar which an agent can have is a context free or regular grammar here.

Communication. Agents communicate with each other by speaking and recognizing *words*, each composed of a finite string of symbols.

All agents derive words using their own rewriting rules. To derive a word a leftmost symbol equal to the left-hand side of a rewriting rule α is rewritten by the right-hand of the rule β. Derivation always starts with an initial symbol S. If a agent has more than two fitting rules in its rule set, the agent adopts one rule randomly. When no more nonterminal symbols are left in the derived word, a derivation terminates. And the derived word is spoken to all agents. An agent fails to speak a word when (i) the derivation does not finish within 60 rewriting steps or (ii) there is no applicable rule its rule set. The length of a word w (denoted by $|w|$) is given by the number of symbols in it. The possible largest length of a given word is M. The words longer than M are truncated after M-th symbol and then are spoken. The possible number of words (N_{all}) is limited to $2^{M+1} - 2$, and a full set of words speakable by an agent G_i is denoted by $L_{\text{sp}}(G_i)$.

Agents try to recognize words by applying their own rules in the opposite direction. If an agent can rewrite a given word back to the symbol S within 500 rewriting steps, we say that the agent can recognize the word. The language recognized by an agent G_i is denoted by $L_{\text{rec}}(G_i)$. Note that the inclusion relationship (i.e. $L_{\text{sp}}(G_i) \supseteq L_{\text{rec}}(G_i)$) holds, because of the truncation and limitation of rewriting steps.

2.2 Communication Game and Evolutionary Dynamics

We set a communication game in a network consisting of P agents. Each agent speaks in turn and each word is given to all the agents. Then every agent including the speaker tries to recognize the word. One time step consists of R rounds. For each round, every agent has a opportunity to speak.

Score. Each agent is ranked by three scores; speaking, recognizing and being recognized. A word spoken by the l-th agent to the m-th agent at a round c is denoted by $w_{lm}(c)$. The scores at round c is computed as follows:

For the factor of speaking, a score is given by

$$p_l^{\mathrm{sp}}(c) = \begin{cases} |w_{lm}(c)|/(trend+1) & \text{, for speaking a word } w_{lm}(c) \\ -3 \ , & \text{for failing to speak any word} \ , \end{cases} \tag{2}$$

where *trend* is defined as the frequency of the word spoken in the last 10 time steps. An agent gets a higher value of $p_l^{\mathrm{sp}}(c)$ when he speaks a longer word and/or a less frequent word.

For the factor of recognition, a score is given by

$$p_{kl}^{\mathrm{rec}}(c) = \begin{cases} |w_{kl}(c)|/s \ \text{, for recognizing a word spoken by } k\text{-th agent} \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{in } s \text{ rewriting steps} \\ -|w_{kl}(c)| \ \text{, for not recognizing a word spoken by } k\text{-th agent} \ . \end{cases} \tag{3}$$

A quick recognition of a long word provides a higher value of $p_k^{\mathrm{rec}}l(c)$.

For the factor of being recognized, a score is given by

$$p_{lm}^{\mathrm{br}}(c) = \begin{cases} |w_{lm}(c)|/P \ , & \text{if the spoken word is recognized by an by } l\text{-th agent} \\ -|w_{lm}(c)|/P \ , & \text{if the spoken word isn't recognized by } l\text{-th agent} \ . \end{cases} \tag{4}$$

Mutually recognizing agents will have high p_{kl}^{br}.

The total score in a time step is an average of a weighted sum of the three scores over R rounds:

$$p_l^{\mathrm{tot}} = \frac{1}{R}\sum_{c=1}^{R}(r_{\mathrm{sp}}p_l^{\mathrm{sp}}(c) + r_{\mathrm{rec}}\sum_{m=1}^{P}p_{lm}^{\mathrm{rec}}(c) + r_{\mathrm{br}}\sum_{k=1}^{P}p_{kl}^{\mathrm{br}}(c)) \ , \tag{5}$$

where r_{sp}, r_{rec} and r_{br} are the respective weighting coefficients. For example, if r_{br} is given a positive value, those who can be recognized by more agents get higher scores. But if the value is set negative, being recognized is no more favorable.

Mutations. In each time step, new agents are produced. The rule set of new agent is inherited from its ancestor's and suffers one of the following three mutation processes: a) *adding mutation* – a new rule is added, which is a modified rule of randomly selected from an ancestor's rule set. b) *replacing mutation* – a

randomly selected rule is replaced with its modified rule. c) *deleting mutation* – a randomly selected rule is deleted.

The modifications of the selected rule are caused by, (i) replacing a symbol of the left-hand with the other nonterminal symbol, (ii) replacing a symbol in the right-hand with the other nonterminal or terminal symbol, (iii) inserting a symbol in the right-hand side or (iv) deleting a symbol from the right-hand.

Adding mutation is applied to agents within the rate m_{add}, if their scores exceed the average score. Replacing and deleting mutations are applied to all the agents within the rate of m_{rep} and m_{del}, respectively.

3 Results of Simulation

In this paper, a network consists of 10 agents ($P = 10$) and each agent tries to speak 10 times in each time step ($R = 10$). The score of the communication game is computed with the fixed parameters: $r_{\mathrm{sp}} = 3.0, r_{\mathrm{rec}} = 1.0$ and $r_{\mathrm{br}} = -2.0$. Note that agents which can speak less acceptable words are benefited for a negative value of r_{br}. It is expected that a variety of the words in a population will increase. All the mutation rates are set at equal value 0.04 ($m_{\mathrm{add}} = m_{\mathrm{rep}} = m_{\mathrm{del}} = 0.04$). The maximum length of a word is limited to 6 ($M = 6$), therefore the number of possible words N_{all} is 126.

Initially, all agents assumed to have the simplest grammar, i.e. a single rule with one symbol in the both hand side. They are classified as type 3 grammars due to Chomsky's hierarchy. At least, either a rule $S \rightarrow 0$ or a rule $S \rightarrow 1$ should be included to derive words.

3.1 Algorithmic Evolution

We find that evolution of grammar system is accelerated by the characteristic factors, one is a module type evolution and the other one is a loop evolution. Computational ability of an agent is measured by the ratio of recognizable words to the total number of possible words, i.e.

$$\text{Computational ability} = \frac{N(L_{\mathrm{rec}}(G_i))}{N_{\mathrm{all}}} \ , \tag{6}$$

where $N(L_{\mathrm{rec}}(G_i))$ is the number of words which the agent G_i can recognize. Fig. **??** represents the example of evolution of the computational ability from the initial network. The computational ability, as well as the number of the distinct words spoken in the network, we call a *variety* of words, evolves with time.

A tree that displays the derivation path of a given word is called a derivation tree of the word. We put all possible derivation tree of a grammar system in a directed, connected graph. A structure of the graph expresses the algorithm of the grammar. Algorithmic evolution can be seen in the topological changes of this graph.

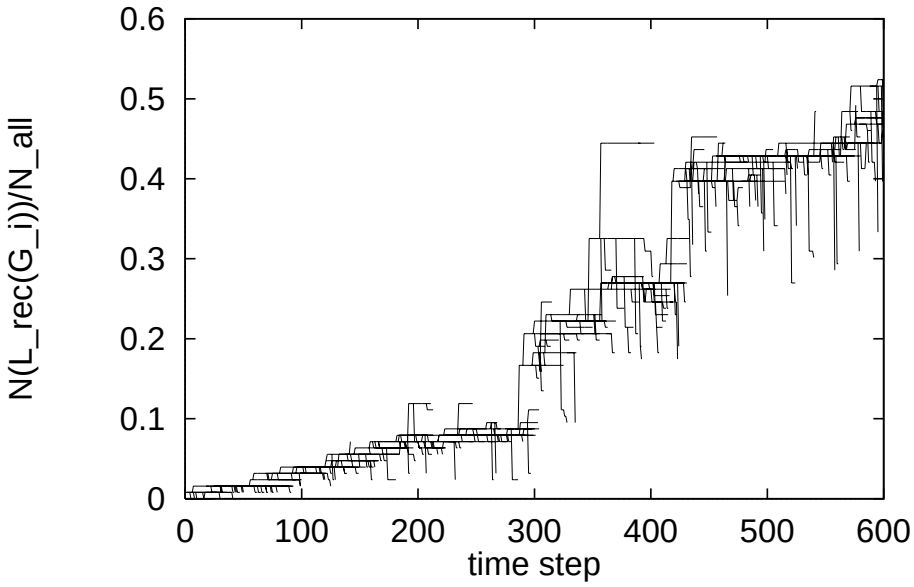

Fig. 1. Time step v.s. $N(L_{\rm rec}(G_i))/N_{\rm all}$. Each line connects one agent to oneself or its offsprings. It branches off by the mutations. A line terminates when the corresponding agent is removed. These lines show upward trend. In initial 200 time steps, computational ability gradually increases. After that, transitions to higher computational agent are frequently observed.

Evolution in Initial Period. It is shown in Fig. **??** that computational ability of agents slowly evolves during initial 200 time steps. In Fig. **??** (a)$\sim$(c) the corresponding grammar systems are depicted in graph diagram. The initial agent has a weakest ability, having a direct derivation rule $S\rightarrow 0$ (Fig. **??**(a)). The agent can increase the ability by the process of the adding mutation. Adding the rule $S\rightarrow 1$ to a production graph generates a branch structure (Fig. **??**(b)). Further, the adding mutation evolves the multi branch structure (Fig. **??**(c)).

Module Type Evolution. We find in Fig. **??** that an agent with the remarkably high ability (> 0.1) appears at time step 192. The change of grammar at this time step is sketched in Fig. **??**(d). An acquired rule $A \rightarrow 00$ can double the acceptable size of the word of a grammar. Every intermediate word containing a symbol A can be rewritten by the rule $A \rightarrow 00$. In the sense that one common rule is used by many different words, we call the key rule a module rule. Evolution processes driven by module rules are called module type evolutions.

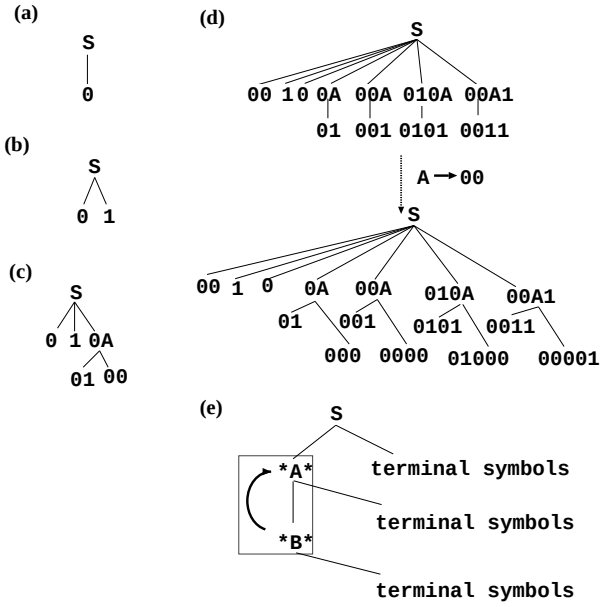

Fig. 2. The examples of grammar structure are shown by graph diagrams: (a) a sequential structure, (b) a branch structure and (c) a multi branch structure. In the (d) An example of module type evolution is shown. Acquiring a rule $A \rightarrow 00$, a grammar without bifurcation (upper tree) is evolved into one with bifurcated branches (lower tree). an example of grammar having a loop structure is schematized in (e). Asterisk stands for any symbols. With this grammar, new agent can rewrite words $*A*$ into $*B*$ and vise versa. Words derived from such grammar can not represented in a tree form.

Emergence of Loop Structure. Grammar systems evolved by modules are not evolutionary stable in general. It is overpowered by more powerful grammar systems. Fig. **??** shows that a new agent with more powerful grammar appears in the population around time step 310. This agent has a loop structure in its grammar system (see Fig. **??**(e)). A loop structure can derive a potentially infinite numbers of words recursively. A grammar with a loop structure is categorized as type 2 grammar or higher one in Chomsky's hierarchy.

3.2 Forming Ensemble with Common Word

An upper structure, which is named an *Ensemble with Common Word (ECW)*, emerges in the population of agents. The ensemble consisted of agents which can speak and recognize the common words. The other agents which can't speak or recognize the common words are less benefited than those in ECW.

When there exists ECW, even an agent of a high ability in a population will

die out. At time step 403, an agent with the highest computational ability of the population dies out (see Fig. **??**). Agents taking too much rewriting steps to recognize words decrease the fitness. We indicate this fact by Table **??**. The rewriting steps to recognize the words at time step 400 are shown in this table.

Agents which cannot recognize frequent words in the population will be removed in order. An agent G_{306} (agent with ID 306), which has the second highest ability in the population, is removed first at time step 400. An agent G_{276} , which has the highest ability in the population, is removed at time step 403. It cannot recognize the word "00". To stay in the population, where a word "00" is the most commonly spoken, each agent should speak and recognize this word quickly. An ability to speak a certain frequent word quickly should be balanced with an ability to speak many words.

Numbers in bold font in Table. **??** represent the largest two rewriting steps to understand the words in the leftmost column. It is clear from this table that it takes much more time for agents G_{306} and G_{276} than the rest of agents. To take more steps to recognize commonly spoken words is disadvantageous for the agents G_{306} and G_{276}. If a group containing agents G_{306} and G_{276} constituted the majority and the words as "001011" or "010101" were the commonly spoken words, agents G_{306} and G_{276} will take an advantage.

Fig. **??** shows the phylogeny of agents at time step 400. It shows that the group consists of the agents G_{276} and G_{306} and that of the other agents forms the different lines. They form two different ECWs. The agents in the major ECW have lower computational ability than those in the minor ECW. Two ECWs conflict to survive in the network. Those in the major ECW behave cooperatively as the result by speaking and recognizing common words and get higher scores. At last all agent in the minor ECW is removed from the network. In this way the evolution toward the high computational ability is suppressed by forming ECW.

After removing agents G_{306} and G_{276} from a network, agents come to compete with each other within the same ECW. Proportional to the number of rewriting steps to recognize the commonly spoken words, the agents are removed from the network. In the ECW, a new agent with the high computational ability will emerge through algorithmic evolution.

3.3 Minimum All Mighty

We can make a *Minimal All Mighty* agent. It is an agent which can speak and recognize all possible words with the least number of rules. For example, a Minimal All Mighty agent has the rules such as:

$$S \to A, A \to SS, S \to 0, S \to 1 \ . \tag{7}$$

This grammar is categorized as a type 2 grammar. It recognizes all the words very quickly and speaks all the words. However, it shows a low variety of words because of random adoption from plural fitting rules. A Minimal All Mighty agent cannot invade into the system composed of ECW because of lower variety

Table 1. This table shows rewriting steps to recognize words (the left most column) spoken at time step 400. Simulation parameters are $r_{\mathrm{sp}} = 3.0, r_{\mathrm{rec}} = 1.0 and r_{\mathrm{br}} = -2.0$. In the second column, the trend of each word (frequency of words in the last 10 time steps) is written. Numerals in the first row are ID of each agent at time step 400 in order that the earlier a agent is removed the lefter it is located. If the agent can't recognize the word, no numerals is put. Bold numerals represent the first two longest steps among agents.

word	trend	**306**	302	307	**276**	305	301	299	294	290	284
001001	30			121	**185**			121	**145**	133	121
011001	20			157	**423**			166	**214**	190	157
11100	16			52	**245**			46	**58**	52	52
11	14		**7**		**12**	**7**	**7**				
110	12		13		**32**	**14**	**14**				
00110	26		53	**57**		**62**	48	43	54	49	53
110010	11		**174**	147				121	**202**	160	147
00	69	**3**	**3**	**3**		**3**	**3**	**3**	**3**	**3**	**3**
10	57	**7**		5	**7**	5	5		5	5	5
011010	24	**431**	210	164	**431**	120	180	166	251	209	164
001010	31	**233**	134	124	**236**	106	106	116	161	141	124
1110	24	**125**	45	27	**122**	60	60	24	29	27	27
01110	9	**122**	91	69	**192**	55	76	66	83	75	69
00101	25	**91**	65	58	**91**	52	49	57	66	63	58
111	30	**53**	21	13	**52**	24	24	13	13	13	13
0001	78	**20**	19		**20**	21	16	18	18	18	17
001011	14	**401**			**404**						
010101	10	**190**			**193**						

of words. In the case of $r_{\mathrm{rec}} = 1.0, r_{\mathrm{sp}} = r_{\mathrm{br}} = 0.0$, an agent whose grammar contains the rules same as rules in (**??**) has evolved. For short point such as low variety of speaking words has no effect on its fitness in this setting.

3.4 Punctuated Equilibrium

We have seen that our system shows rapid algorithmic evolution of the grammar systems in certain stages. On the other hand, algorithmic evolution is suppressed by forming ECW. Rapid algorithmic evolution follows quasi-equilibrium stages. Temporal evolution of amounts of handling information therefore shows punctuated equilibrium phenomena (Fig **??**). Because handling information defined below is sensitive to the formation of ECW.

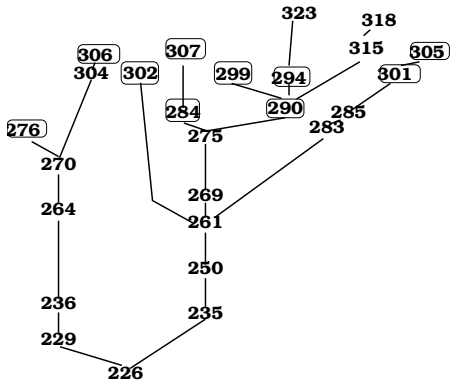

Fig. 3. This picture represents the phylogeny of agents at time step 400 (in oval boxes) from common ancestor (G_{226}). A number represents ID of each agent. A line is drawn from parent agent (lower) to its offsprings (upper). Two genetic series are bifurcated from the common root (G_{226}), the agents G_{306} and G_{276} and that of the other agents. They are forming different ECW. The agent G_{306} and G_{276} are both contained in the left series.

The handling information of the l-th agent is defined as the follows,

$$f_l = \frac{1}{RP^2} \sum_{c=1}^{R} \sum_{k=1}^{P} |w_{kl}^{\mathrm{rec}}(c-1)| \sum_{m=1}^{P} |w_{lm}^{\mathrm{rec}}(c)| \tag{8}$$

$$|w_{ij}^{\mathrm{rec}}(x)| = \begin{cases} 1 & \text{if } x = 0 \\ \text{the length of } w_{lm}(x) & \text{if } x > 0 \text{ and the word which spoken by the} \\ & \quad i\text{-th agent is recognized by the } j\text{-th agent} \\ 0 & \text{otherwise .} \end{cases} \tag{9}$$

Information contents of a word is simply given by the length of the word as the first approximation. The initial amount of handling information, i.e. $|w_{ij}(0)|$, is defined as 1.

If an agent gets high f_l value, which suggests that the agent can recognize words spoken by the others and the agent's speaking words are recognized by the other agents. When some ECWs conflict with the other ECWs, the averaged handling information in a population, $\langle f_l \rangle = \sum_{l=0}^{P} f_l/P$, does not increase. After some ECWs occupy the whole network, long and new words will again be spoken to and recognized by the agents. Punctuated equilibrium phenomena in the amount of $\langle f_l \rangle$ is explained by the scenario.

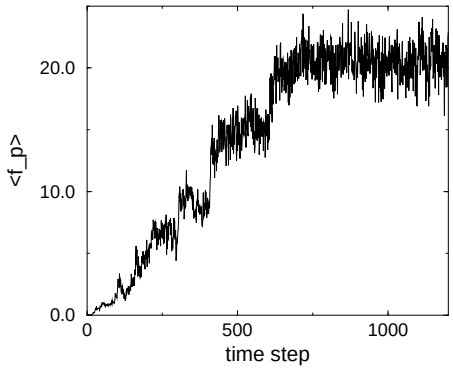

Fig. 4. time step v.s. the average handling information (see the definition in the text): In the first 700 time steps, evolution can be observed. The stepwise changes reflect alternate evolution of ECW and the algorithmic evolution.

4 Summary and Discussion

We have studied the evolution of symbolic grammar, by introducing a network model of communicating agents. Each agent has its own grammar system, being expressed as a set of rewriting rules. Via a combination of rules, each agent speaks words to the other agents and tries to recognize words spoken by the other agents. When mutationally dynamics of grammars are introduced, agents' grammar systems change in time. Generally, an agent can speak and recognize more words if it has more rules. Hence agents with more rules can breed more. However the number of recognizable words is not a simple function of the number of rules.

In the present paper, we have shown that two processes are significant in evolution of grammar systems. One is a module type evolution. If a rule becomes a module, which means that it can be utilized by many words to generate nearly twice as many words as before. The number of recognizable words rapidly increases when this module emerges in a grammar system. The other one is a loop forming evolution. The grammar possessing a loop structure can derive recursively many words. It should be noted that a grammar with a loop structure cannot be represented in a tree shape. Namely, the grammar system climbs up Chomsky's hierarchy by one step from type 3 to type 2. Hence we regard such loop forming evolution as a single algorithmic evolution.

It is often believed that a grammar system in the higher hierarchy will perform better than grammars in the lower hierarchy. It may be argued that agents sharing the highest grammar system will dominate in population sooner or later. This argument would be valid if we evolved a grammar system without a network structure.

But this is not necessarily so for agents forming communication network.

Synergetic behavior of agents generates a macro structure named ECW (Ensemble with Common Words). Within this ensemble, several words are spoken and recognized in common. In order to speak and recognize the common words more quickly, it is wiser to have the words as single rules (i.e. $S \rightarrow$ words). This becomes a restrictive condition for individual grammar systems. That is, any agent to survive in the ensemble has to evolve its rule set within this restriction.

This restrictive condition imposed by the ensemble structure disturbs the smooth evolution to the highest ability grammar systems. Therefore all mighty agent, i.e. it can speak and recognize all possible words, is difficult to emerge. The restriction and the algorithmic evolution occur in order, the average handling information shows a punctuated equilibrium evolution.

It is interesting to note that this restriction is not given as an individual rule, but spontaneously emerges from the evolution course of a network. The emergent restriction on each grammar system can be called a net grammar. If we define grammar as meta rule sets that give restrictions on a possible language set, this ensemble structure can be the example of such grammar. We have succeeded in showing the meta grammar structure and dynamics conducive to understand an actual language system.

Acknowledgments

The authors would like to thank N.Nishimura for critical reading of the manuscript. They thanks Eken S.Yoshikawa for stimulating discussions. One of the authors (T.H.) wish to express his gratitude to T.Yamamoto and N.Matsuo for helpful comments.

References

1. Hopcroft, John E. and Ullman, Jeffrey D.: *Introduction to Automata Theory, Languages and Computation* Addison-Wesley Publishing Co,, 1979.
2. Révész, György E.: *Introduction to Formal Languages* Dover Publications, Inc., 1991.
3. MacLennan, Bruce: Synthetic Ethology: An Approach to the Study of Communication in *Artificial Life II* Addison-Wesley Publishing Co,, 1992.
4. Werner, Gregory M. and Dyer, Michael G.: Evolution of Communication in Artificial Organisms in *Artificial Life II* Addison-Wesley Publishing Co,, 1992.
5. Lindgren, Kristian: Evolutionary Phenomena in Simple Dynamics in *Artificial Life II* Addison-Wesley Publishing Co,, 1992.
6. Ikegami, Takashi: From genetic evolution to emergence of game strategies Physica D **75** (1994) 310 – 327
7. Ray, Thomas S.: An Approach to the Synthesis of Life in *Artificial Life II* Addison-Wesley Publishing Co,, 1992.
8. Ray, Thomas S.: Evolution, complexity, entropy and and artificial reality Physica D **75** (1994) 239–263

This article was processed using the LaTeX macro package with LLNCS style