

Growing Self-Organizing Map Approach for Semantic Acquisition Modeling

Mengxue Cao*, Aijun Li*, Qiang Fang* and Bernd J. Kröger^{†‡}

*Laboratory of Phonetics and Speech Sciences, Institute of Linguistics
Chinese Academy of Social Sciences, Beijing, China

Email: mengxuecao@outlook.com, liaj@cass.org.cn, fangqiang@cass.org.cn

[†]Neurophonetics Group, Department of Phoniatics, Pedaudiology, and Communication Disorders, Medical School
RWTH Aachen University, Aachen, Germany

[‡]Cognitive Computation and Applications Lab., School of Computer Science and Technology
Tianjin University, Tianjin, China

Email: bkroeger@ukaachen.de

Abstract—Based on the incremental nature of knowledge learning, in this study a growing self-organizing neural network approach for modeling the acquisition process of semantic features is proposed. The Growing Self-Organizing Map (GSOM) algorithm is extended and applied to the problem of language acquisition. Based on that algorithm, experiments are conducted using Standard German children’s books corpus. A cyclic reinforcing and reviewing training procedure is introduced to model the teaching and learning process between children and their communication partners. Experimental results indicate that (1) GSOM has good ability to learn the semantic categories presented within the training data, that (2) clear semantic boundaries can be found in the network representation, and that (3) cyclic reinforcing and reviewing training leads to a detailed categorization of lexical items as well as to a detailed clustering, while keeping the already learned clusters and already developed network structure stable. Experiments show that our GSOM approach is a good method for modeling semantic learning during language acquisition.

Keywords—Neural network, growing self-organizing map, semantic feature map, language acquisition

I. INTRODUCTION

During language acquisition, infants face the task of learning various kinds of information and of organizing that information into linguistic categories. During that process, however, infants do not receive explicit language instructions, nor are they able to make inquiries about the structure that they are learning [1]. Instead, they must discover the linguistic categories of their native language from their interactions with communication partners. The task is further complicated by the fact that they do not know how many categories to discover along any particular input dimension [1]. The appropriate underlying learning mechanisms for infants, however, are still unclear. In this study we propose a feasible approach that it is capable to explain the acquisition process of semantic categories. That approach can contribute to the merging process of *Information and Communication Technologies* [3] and *Cognitive Infocommunications* [2] to some extent by modeling the teaching and learning process between infants and their communication partners.

Kohonen [4] proposed a kind of self-organizing neural network, known as the Self-Organizing Map (SOM), which

has the ability to project high-dimensional data onto a two-dimensional feature map. Its high visualizing feature enables the analyst to overview the category structures of a data set. Later, Ritter and Kohonen applied the SOM algorithm onto semantic tasks [5]. Their research revealed that SOM has the ability to detect the “logical similarity” between words and group similar words into clusters.

In recent years, the topographic perceiving feature and the self-organizing ability of SOM have been applied increasingly for tasks on modeling the human acquisition process. In linguistic field, for example, Kröger *et al.* [6] modeled the acquisition of vowel and consonant categories using SOM, Zinszer *et al.* [7] developed a SOM model which models the first language lexical attrition phenomenon, and Warlaumont *et al.* [8] investigated the reinforcement effect on vocal motor learning based on SOM.

Although SOM is reasonable for modeling the topographic structure and the knowledge reorganization of a learning process, it does have limitations in modeling the incremental nature of knowledge growth. Due to the phenomenon of catastrophic interference [9], SOM has difficulties in adapting new knowledge into an existing trained network. In other words, the structure of SOM cannot be extended easily, thus cannot be directly used to model the knowledge learning process realistically.

Many researchers made an effort aiming to overcome that problem, such as the Growing Cell Structures (GCS) proposed by Fritzke [10] and the Neural Gas Algorithm (NGA) proposed by Martinetz and Shulten [11]. However, GCS has limitations in representing high dimensional data, while NGA is restricted by its limited map size. Those algorithms, therefore, are not suitable for modeling linguistic categories.

Exploring extendable SOM in data mining field, Alahakoon *et al.* [12] proposed an extendable version of SOM called the Growing Self-Organizing Map (GSOM), which let new nodes smoothly join the existing network and dynamically extend the size of the network. Its dynamic structure was proved to be very effective for knowledge discovery applications.

In this paper, that Growing Self-Organizing Map algorithm is adapted for language acquisition modeling.

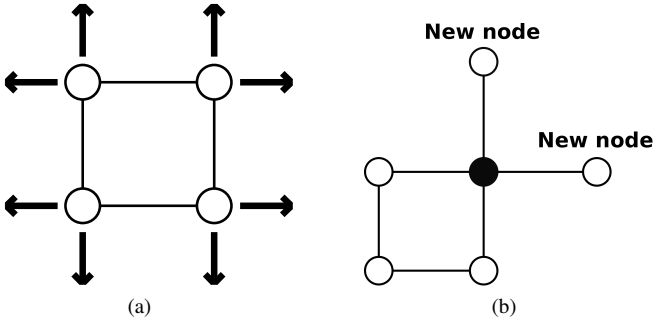


Fig. 1: The initial structure of GSOM. (a) The network can be expanded to any direction at the beginning. (b) New nodes can grow the network at boundary nodes. [12]

II. GROWING SELF-ORGANIZING MAP

A. Structure of Growing Self-Organizing Map

Compared with traditional SOM, the structure of Growing Self-Organizing Map (GSOM) is simpler. Instead of having a size-predetermined rectangular map, the network of GSOM does not have a fixed size or shape. Starting from 4 initial nodes, new nodes can grow at boundary nodes and smoothly join the existing network (*see* Fig. 1b). Thus, the network can be dynamically expanded to any direction outwards depending on the new growing nodes (*see* Fig. 1a).

Two factors, the accumulative error (AE) and growth threshold (GT), are introduced into GSOM. The error value is calculated by the Euclidean distance between an input vector and the weight vector of the best matching unit (BMU). Thus, each BMU node has an error value as an additional characteristic parameter, and its value is accumulated throughout the training process. When the AE value of a BMU exceeds the value of GT , that corresponding Voronoi region [13] is said to be underrepresented, and new nodes are then introduced to the network. A high GT value will result in a less spread map, while a low GT value will produce a well-spread map [12].

B. Basic Training Process

Training data (i.e. A set of N training items consisting of feature vectors) are treated as input tokens for the network training process. The basic training process of Growing Self-Organizing Map contains 2 phases:

1) *Initializing Phase*: At the beginning, 4 neurons are initialized. Their weight vectors are assigned randomly within interval 0 and 1, and their AE values are initialized to 0. Instead of stated in [12], in our approach, GT is set arbitrary to fit our experimental requirements.

After initialization, all starting nodes are boundary nodes and thus are free to grow in any direction outwards. That results in great flexibility concerning network growth.

2) Growing Phase:

a): Input tokens are presented to the network one by one sequentially. Each token will be trained for several times before next token enters. That is consistent with the gradual learning process in practice: parents often teach their children just one word at a time, and repeat it for several times.

b): In GSOM, the weight update and network reorganization processes are performed locally. Therefore, learning rate and neighborhood size are initialized to their initial value with each new input token. The learning rate is defined as a function of the total number of nodes in the network by (1), where α is the reduction factor of learning rate with $0 < \alpha < 1$; $\varphi(n) = 1 - R/n(t)$ is a function of the total number of nodes in current network; R is a constant set to 3.8 as in [12]; and $n(t)$ is the number of nodes in the network at time t .

$$LR(t+1) = \alpha \times \varphi(n) \times LR(t) \quad (1)$$

c): The distances between weight vectors and training vectors are calculated using Euclidean distance measure, and the best matching unit (BMU) with minimum distance is detected within the network for the current training token.

d): If the AE value of this BMU exceeds GT and this BMU is a boundary node, then new nodes grow at all free direct neighboring positions. Then weight vectors of new nodes are initialized with regard to the weight vectors of this BMU and its neighbors (that process is called *weight distribution*).

e): If the AE value of this BMU does not exceed GT , then weight update is applied to this BMU and its neighbors within the neighborhood. Gaussian distribution is chosen as a part of the neighborhood function that can be represented by $h(t) = \beta \times \exp(-d_{ix}^2/2\sigma_t^2)$, where β is the reduction factor of neighborhood size with $0 < \beta < 1$; d_{ix} represents the distance between weight vector i in BMU and training vector x ; σ_t represents the current neighborhood size. The weight update function can be expressed by (2). The Euclidean distance between this BMU vector and the training vector is accumulated as the AE value of this BMU.

$$\omega_i(t+1) = \omega_i(t) + LR(t) \times h(t) \times (x(t) - \omega_i(t)), i \in N \quad (2)$$

f): If the AE value of this BMU exceeds GT but this BMU is not a boundary node, then *error distribution* is performed. Then, the error value of this BMU is reduced to $GT/2$, and the error values of its immediate neighbors are increased by γGT ($0 < \gamma < 1$).

g): Then several iterations are done for the current training token (go through steps *c*) to *f*) several times). The learning rate and neighborhood size decrease at each iteration. The iteration process stops when neighborhood size reduces to unity.

h): Then, the next training token is processed by repeating steps *b*) to *g*) until all training tokens are presented.

C. Checking Process

The checking process does not change the network. It is performed to check whether the trained network has learned a good representation of categories represented within the training data, by identifying the winner positions in the trained network for each token. That can be considered as a calibration phase if known data are used [12]. The closeness of each token to each neuron in the network is measured by Euclidean distance.

D. Reinforcing and Reviewing Training

During language learning process, children cannot learn the whole knowledge (i.e. all semantic categories represented by the training tokens) at once, so that imperfections in clustering are inevitable. During semantic acquisition, those errors could represent a current incapability of distinguishing words with different meanings. Reflected in the network, that fact is represented by those neurons which represent many words after performing the checking process. This is comparable to the following natural learning situation: when parents teach their children, and if they find that some words are always confused by their children, parents will repeat those words and reinforce the differences between those words, in order to help their children get them distinguished. During that reinforcement process, some learned words will also show up in the communication between parents and children.

When growing phase is completed and checking is performed, the trained network may end up with some “unsolved” edge nodes, which means some nodes may represent the characteristics of many tokens comprising different semantic categories. In order to resolve those “high-density” nodes, a series of additional reinforcing and reviewing training can be performed.

During the reinforcing phase, the trained network from growing phase is the starting network. The training process is similar to that in the growing phase. Training data are consisted of those words represented by “high-density” neurons in the checking results. The initial learning rate is increased to give more weights to the input token, and the *GT* value is decreased to stimulate the network growth at “high-density” nodes.

During the reviewing phase, the trained network from current reinforcing phase is the starting network. The training process is similar to that in the growing phase. Compared with reinforcing phase, more tokens from the training set are used in the reviewing process to simulate the reoccurring of learned words. The initial learning rate and the *GT* value are set to the same as those in the growing phase.

A reinforcing phase followed by a reviewing phase together forms a combined training process of reinforcing and reviewing training. According to our experimental requirements, this combined training process repeats in a cyclic (or iterative) way for several times.

III. EXPERIMENT AND RESULTS

A. Standard German Children’s Books Corpus

In this study, the Standard German children’s books corpus [14] was used as the basis for generating our training set. The Standard German children’s books corpus comprises transcriptions of 40 books targeted to children from age 1 to 6. In total, 6513 sentences and 70512 words are transcribed in this corpus. Morphologically distinct forms of the same word are counted as separate words (e.g. “Blume” meaning *flower* and “Blumen” meaning *flowers*, are treated as two different words). The corpus therefore consists of 8217 different words, which is assumed to approximately represent a 6-years-old child’s mental lexicon. In this study, only nouns were used as training data, and only the first 332 most frequent nouns were chosen. The top ten frequent nouns are listed in TABLE I.

TABLE I: Top ten frequent nouns in the 332 word data set

Frequency counts in corpus	Frequency counts in training data	German words	English translation
392	78	Mama	mom
278	55	Bär	bear
235	47	Papa	dad
217	43	Mond	moon
190	38	Kinder	children
147	29	Katze	cat
145	29	Frau	wife
106	21	Bett	bed
105	21	Mädchen	girl
104	20	Wasser	water

B. Training Data Set

Two native speakers of Standard German (undergraduate students from RWTH Aachen University) developed a list of semantic features for the corpus by a simple brain storming procedure. In total, 1715 features were developed. To reduce dimensions of the training vectors, we dropped features which only occur once, while keeping all words distinguishable by their semantic features. Finally, 724 features were kept (the top ten features are listed in TABLE II). Therefore, each word in our training data set is represented by 724 feature vectors. Binary coding is used for the representation of each word. Thus, for each word, among its 724 vectors, “1” is used to mark its own features and “0” is used to mark features not belong to this word.

To model the semantic acquisition process, the frequency of word occurrence was taken into account when building the training set. The words were presented proportional to the frequency of occurrence of a word in the corpus (*see* the second column of TABLE I).

The 332 words arranged in sequence by descending order of their frequency counts constitute the basis of the training set. Based on that, repeated words were randomly inserted to the 332-word list regarding to the frequency counts of each word. To keep the frequency order, we exclusively allowed higher frequent words to be inserted after lower frequent words. For example, the first rank is “Mama” and then “Bär”. After “Bär”, “Mama” has 50% possibility to be inserted. Then the next rank is “Papa”. After “Papa”, “Mama” and “Bär” each has 50% possibility to be inserted. The insertion of certain words would be stopped when words reached their frequency counts. In total, 1929 word tokens comprised our training data set.

C. Experiment Procedure and Results

As introduced in Section II-B and Section II-D basic *growing training* and *reinforcing and reviewing training* experiments were conducted respectively. The experiment process was divided into 21 training steps. Step 1 represents the basic growing training. The following 20 steps represent the steps in the cyclic reinforcing and reviewing training (from step 2, even numbers represent reinforcing training steps and odd numbers represent reviewing training steps). In total, one basic growing training and 10 cycles (20 steps) of reinforcing and reviewing training were performed in this experiment. The 1929-word

TABLE II: Top ten semantic features in the 332 word data set

Frequency	German features	English translation
81	ist ein Gegenstand	is an object
63	hat zwei Augen	has two eyes
60	es gibt verschiedene Arten	there are different types
56	hat einen Kopf	has a head
47	hat eine Nase	has a nose
46	hat zwei Beine	has two legs
42	hat eine Haut	has a skin
37	ist aus Kunststoff	is made of plastic
37	ist aus Metall	is made of metal
36	ist ein Mensch	is a human

training set was used as the main training data, and the 332-word list (*see* Section III-B) was used as the testing set.

1) *Basic growing training*: First, a growing training was performed based on the training set of 1929 training tokens with 724 feature vectors in each. After training, a checking process was done by inputting the 332-word list with their feature vectors to the trained network, and winner neurons were found. The trained network structure and the checking result are shown in Fig. 2.

From Fig. 2, general clusters of semantic categories can be found, such as “Persons”, “Animals”, “Clothes”, “Transportation”, “Housewares”, “Snacks” and “Body parts”. That proves GSOM has the ability to learn the semantic categories in the training data and build semantic clusters even with such high dimensional training vectors. A separation between “*Living*” and “*Dead*” can be seen that “*Living*” things are located on the top-left and down-right part, while “*Dead*” things are located through the middle part.

In the network, some neurons represent more than one word. Taking a closer look at the position of those neurons, we find most of them are located at the edge areas of the network. With proper reinforcing and reviewing training, we expected those neurons at network edge areas could stimulate the network to grow further in that region in order to resolve those words represented by a “high-density” neuron.

2) *Cyclic reinforcing and reviewing training*: Based on the current trained network, a cyclic reinforcing and reviewing training was conducted. Each cycle contains two phases, a reinforcing phase and a reviewing phase. The trained network in each step was used as the starting network of next step.

For the reinforcing phase, in each current trained network, by examining the checking result, if a neuron represents more than 4 words, or the average Euclidean distance between the neuron and its represented words is bigger than 2.5, those corresponding words are then taken as the further training data. For simplicity, no frequency influence was considered in this phase. For example, 218 words were chosen for the first reinforcing training step. The trained network from the growing training was used as the starting network of the first reinforcing training step. The initial learning rate was increased to give more weights to the input token simulating the reinforcement from parents, and the *GT* value was decreased to stimulate the network growth at those “high-density” node regions.

For the reviewing phase, the starting network was the

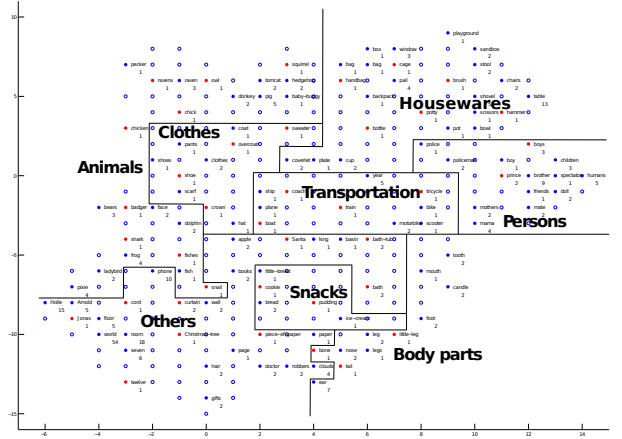


Fig. 2: The trained network structure and checking result of basic growing training. (Filled nodes represent those neurons with word representations; empty nodes represent neurons without word representations. Red nodes represent those neurons whose average Euclidean distance between its represented words and itself is smaller than 0.5; blue nodes represent neurons with average distance bigger than 0.5. Labels beside nodes are the best represented word of that neuron and numbers blow are the number of words represented by that neuron. Annotations in the figure represents different semantic clusters, and solid lines represent cluster boundaries.)

trained network from current reinforcing phase. The 332-word list comprised the training set. The initial learning rate and the the *GT* value were set to the same values as those of growing training in Section III-C1.

After each training step, the same 332-word checking (as already used for the growing training) was performed. The trained network structure and the checking result of the first reinforcing training phase are shown in Fig. 3. From Fig. 3, we can clearly see a growth at the down-left part of the network. Among those neurons, lots of “high-density” neurons have been resolved, and only a few still exist. In addition, more words are represented across the retrained network. Some local reorganization can be noticed during this reinforcement training procedure. However, the learned clusters are clearly kept and no catastrophic interference is caused by the reinforcing training. In addition, some new clusters such as “Time”, “Plants”, “Places”, “Numbers”, “Names” and “Minds” are formed at the expanded parts of the network. Those results indicate that the reinforcing training can help the network distinguish “unsolved” words and build more detailed clusters while keeping the learned network structure stable.

The trained network structure and the checking result after 10 training cycles are shown in Fig. 4. From Fig. 4, we can see that the down-left part of the network in Fig. 3 has developed into a network with comparable size and shape of the upper part. Although very few “high-density” nodes still exist, the whole network are now very well developed. Although some reorganizations happen, the learned clusters and network structure in the network in Fig. 3 are well kept and not destroyed by the cyclic reinforcing and reviewing training.

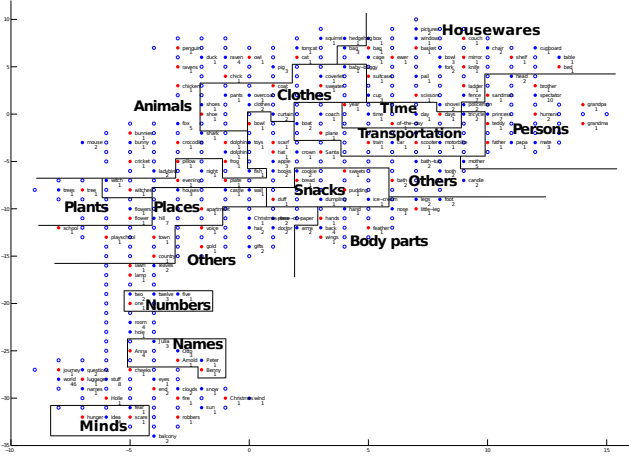


Fig. 3: The trained network structure and checking result of the first reinforcing training phase. (Please refer to Fig. 2 for figure caption)

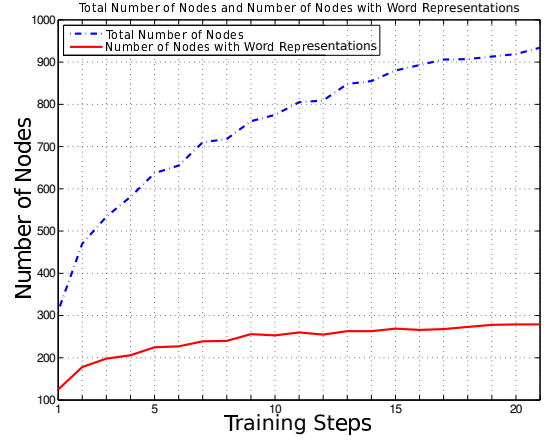


Fig. 5: Total number of nodes and number of nodes with word representations. (blue dotted line represents the total number of nodes, red solid line represents the number of nodes with word representations)

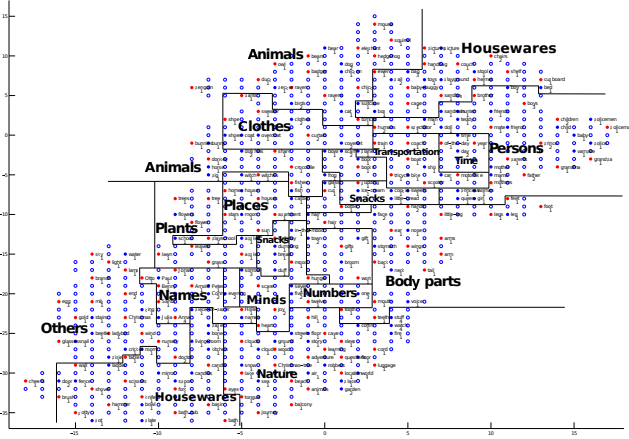


Fig. 4: The trained network structure and checking result after the final step of cyclic reinforcing and reviewing training. (Please refer to Fig. 2 for figure caption)

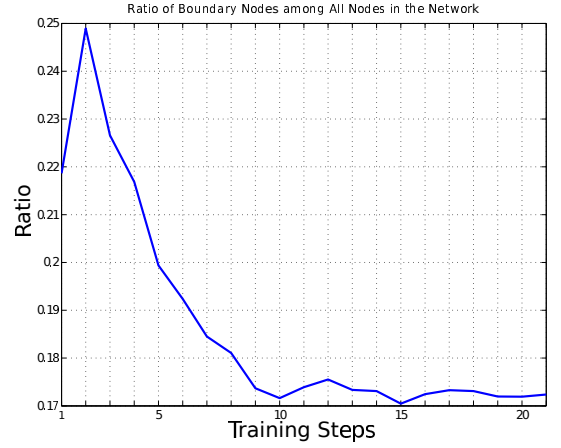


Fig. 6: The ratio of number of boundary nodes over total number of nodes in the network.

IV. PERFORMANCE ANALYSIS

The performance of our GSOM algorithm in modeling semantic acquisition is further analyzed in this section. The analysis is divided into 21 steps to track the training result of each step of the experiment process.

A. Network Expansion

During the training process, the network keeps growing. Fig. 5 shows the growing trends of total number of nodes and the number of nodes with word representations. Both of them increase sharply after the first reinforcing training. The huge peak in Fig. 6 also indicates a significant expansion of a branch network after the first reinforcing training. That increase is consistent with the growth shown in Fig. 3. Then, the network keeps expanding itself, but the increasing rate gradually decreases. The curve in Fig. 6 declines with the training process after the first reinforcing training, and then gradually becomes stable. That reveals the process that the

network occupies another nearby neuron region after the first reinforcing training and gradually forms into a compact network. The final network ends up with a reasonable size of 934 nodes, and a good neuron representation resolution of 279 words (84% of all words are resolved).

B. Semantic Representation

During the training process, the network gets gradually better semantic representation of words. Fig. 7 shows the maximum number of words represented by a single neuron. A generally declining trend can be found with the development of the cyclic reinforcing and reviewing training. Although some fluctuations can be noticed during the middle and late stages of the training process, the average words represented by a neuron in the network decreases continuously as shown in Fig. 8. That means the cyclic reinforcing and reviewing training can help the network resolve “high-density” nodes, thus can help children to disambiguate the meaning of words in the case of clustered words at one node.

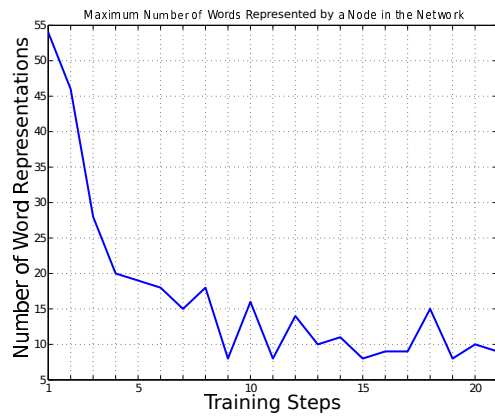


Fig. 7: The maximum number of words represented by a neuron in the network.

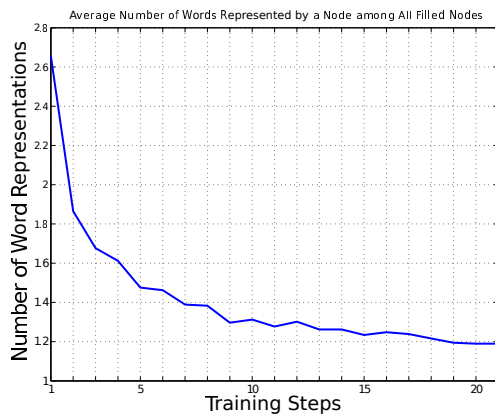


Fig. 8: Average number of words represented by a neuron among all those nodes with word representations.

V. DISCUSSION AND CONCLUSION

In this paper, the Growing Self-Organizing Map algorithm is extended and introduced as an approach to model the acquisition process of semantic features. Its self-organizing ability enables it to model the clustering of semantic categories, and its dynamic growing structure enables it to model the incremental nature of knowledge growth. By reading in inputs one at a time and training it for iterations till the next, the training process of GSOM simulates the gradual learning process in practice. By introducing the GSOM algorithm based on the accumulative error and growth threshold factors, the network can produce a vivid biologic picture of the knowledge learning process. The weight distribution passes the neighborhood structure to the new adding nodes, so that the network structure can keep stable throughout the growing stage. Its learning rate adaptation and localized weight update rules ensure that tokens enter relatively late can also occupy their stable regions in the network.

Based on our GSOM algorithm, the semantic acquisition process was modeled by a cyclic reinforcing and reviewing training. The reinforcing training steps simulate a situation in which communication partners repeat confusing words and reinforce their specific meaning in order to help their children

(the network) to get them distinguished. The reviewing training steps simulate the reoccurring of some already learned words during the reinforcement learning process. From experiment results, our GSOM algorithm in combination with the use of reinforcing and reviewing procedures shows good ability to learn the semantic features in the training data and build semantic clusters even with such high dimensional training vectors. From the checking results, clear semantic boundaries can be found in the network neuron representation. Thus, cyclic reinforcing and reviewing training is proved to be able to help the network distinguish “unsolved” words and build more detailed clusters while keeping the learned clusters and network structure stable.

Although the GSOM is a highly abstract neurocomputational model, it could be carefully interpreted as biologically plausible to a specific degree, because it incorporate important neurofunctional principles like self-organization, associative learning, Hebbian learning, adaptation, and neural plasticity.

ACKNOWLEDGMENT

This research was funded by the NSFC Key Project with No. 61233009, and the CASS Innovation Program.

REFERENCES

- [1] B. Gauthier, R. Shi and Y. Xu, *Learning Phonetic Categories by Tracking Movements*, Cognition, Vol. 103, No. 1, 2007, pp. 80-106.
- [2] P. Baranyi and A. Csapo, *Definition and Synergies of Cognitive Infocommunications*, Acta Polytechnica Hungarica, vol. 9, 2012, pp. 67-83.
- [3] G. Sallai, *The Cradle of the Cognitive Infocommunications*, Acta Polytechnica Hungarica, Vol. 9, No. 1, 2012, pp. 171-181.
- [4] T. Kohonen, *The Self-Organizing Maps*, 3rd Edition, Springer, Berlin, 2001.
- [5] H. Ritter and T. Kohonen, *Self-Organizing Semantic Maps*, Biological Cybernetics, Vol. 61, No. 4, 1989, pp. 241-254.
- [6] B. J. Kröger, J. Kannampuzha and C. Neuschaefer-Rube, *Towards A Neurocomputational Model of Speech Production and Perception*, Speech Communication, Vol. 51, No. 9, 2009, pp. 793-809.
- [7] B. Zinszer and P. Li, *A SOM Model of First Language Attrition*, Proceedings of the 32nd Annual Conference of the Cognitive Science Society, Portland, 11-14 August 2010, pp. 2787-2792.
- [8] A. S. Warlaumont, G. Westermann, E. H. Buder and D. K. Oller, *Prespeech Motor Learning in A Neural Network Using Reinforcement*, Neural Networks, Vol. 38, pp. 64-75, 2013.
- [9] R. M. French, *Catastrophic Forgetting in Connectionist Networks*, Trends in Cognitive Sciences, Vol. 3 No. 4, 1999, pp. 128-135.
- [10] B. Fritzke, *Let it Grow Self Organizing Feature Maps with Problem Dependent Cell Structure*, Proceedings of the International Conference on Artificial Neural Networks, Amsterdam, 1991, pp. 403-408.
- [11] T. M. Martinetz and K. J. Schulten, *A Neural Gas Network Learns Topologies*, Proceedings of the International Conference on Artificial Neural Networks, Amsterdam, 1991, pp. 397-402.
- [12] D. Alahakoon, S. K. Halgamuge and B. Srinivasan, *Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery*, Neural Networks, Vol. 11, No. 3, 2002, pp. 601-614.
- [13] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tesselations, Concepts and Applications of Voronoi Diagrams*, 2nd Edition, Wiley, New York, 2000.
- [14] B. J. Kröger, P. Birkholz, J. Kannampuzha, E. Kaufmann, and C. Neuschaefer-Rube, *Towards the Acquisition of a Sensorimotor Vocal Tract Action Repository within a Neural Model of Speech Processing*, Analysis of Verbal and Nonverbal Communication and Enactment. The Processing Issues, Springer, Berlin, 2011, pp. 287-293.

[This paper was published on CogInfoCom13, 2013]