Artificial Intelligence and Language Acquisition – An Example of A New Approach – Solution to the “WUG-Test” Interpreted within the Artificial Mind System Context

Tetsuya HOYA, Noriko AKIHO-TOYODA (Meikai University)

Abstract: This study proposes a novel connectionist model that explains a solution to Pinker's statement about Berko's “wug-test” in terms of the interactive data processing between the short-term/working memory and long-term memory modules within the artificial mind system. Pinker challenged the Rumelhart and McClleland's pattern association model by enumerating several points where the model failed to simulate the human language faculty. We here demonstrate how the artificial mind system works and simulate the acquisition. This may be able to suggest a new approach to the language acquisition study with a scope of the artificial intelligence (artificial mind), while providing solutions to the problems that Pinker raised.

1 Introduction

The approaches in terms of artificial intelligence (AI) have provided useful tools towards elucidating the various faculties of human language. Then, since we can set up for the condition of the artificial devices to a great extent, whilst assuming more precisely a situation than the case of real human subjects, before performing the actual experiments, not only can we repeat exactly the same experiments but also consistent results are expected to be obtained. In general, it is therefore considered that investigating then comparing the results so obtained with real examples can give another significant insight in the study of language acquisition.

2 Language acquisition by connectionist models

Connectionism, or artificial neural networks (ANNs), is a branch of AI study, and, within the aforementioned principle, the research in the ANN field has also been conducted to simulate the faculty of human language acquisition by developing a model and investigating its behaviour through the extensive use of machinery.

About two decades ago, one of the influential connectionist accounts was proposed by Rumelhart and McClleland; i.e. the so-called ‘pattern association’ model (Rumelhart, et al. 1986). They attempted to explain the inflection of verbs in past tense from the original forms, by using a simple two-layered neural network (or, multi-layered perceptron neural network, MLP-NN). However, their model has been claimed not to be appealing both linguistically and computationally; from a linguistic point of view, the model does not generalise well some novel regular verbs (Pinker 2000; Zalta 2006), whilst within the ANN context it is now well-known that tuning of the network parameters in MLP-NNs (i.e. by the so-called back-propagation algorithm) quite often involves numerical instability-related problems and thus that to make MLP-NNs work reliably for a particular task domain (i.e. even for a simple pattern recognition task) is not a straightforward job; for example, it is generally hard to learn new patterns (or, in other words, incremental learning) without affecting the previously stored knowledge by MLP-NNs, unlike humans.

3 Artificial mind system (AMS)

In the article (Hoya 2005), Hoya proposed the complete holistic model of artificial mind system (AMS). Macroscopically, AMS can be viewed as an input-output system and consists of a total of 14 (loosely-distinct) modules: attention, emotion, explicit and implicit LTM, input: sensation, instinct, intention, intuition, language, perception (i.e. the secondary output), primary output (such as action and endocrine), semantic networks/lexicon, short-term/working memory (STM/WM), and thinking module, the notion of which generally agrees with the psychological ‘modularity principle of mind’ (Fodor 1983; Hobson 1999). Then, he proposed that the respective modules and their mutual data processing in AMS can be represented by using a new form of connectionist model, named kernel memory (KM).

3.1 Kernel memory – a novel connectionist model

As in most of conventional ANN models, kernel memory consists of a set of nodes and the mutual connections (i.e. called 'link weights'). However, the notion of nodes as well as the weighted interconnections in kernel memory is rather different from that of conventional ANNs; the following summarises several important aspects of kernel memory:

i) In kernel memory, a node (i.e. kernel unit) can process not only a single but also multiple input values at a time, depending upon the configuration, and transforms the values into its output value(s) by applying either a linear or non-linear function; if a Gaussian response function is chosen as the non-linear transformation of multiple input values, a node is equivalent to a radial basis function (RBF); it yields a *similarity measure* between the input given and the centre position (as well as the width; that is, an RBF can be regarded as an element of memory system; hence the term 'kernel memory'). Thereby, a single node can by itself perform (locally) pattern matching; if the value given by the similarity measure exceeds a certain threshold, it is regarded that the input pattern matches the data internally stored (this is also concretely justified within the pattern recognition/signal processing context; cf. Hoya 2004; 2005).

ii) The link weight between a pair of kernel units simply represents the *strength* of the connection in between, and activation of one kernel unit may invoke that of the other(s) by its transfer via the link weight, even though there is no actual input to the other corresponding unit(s). Then, unlike ordinary RBF networks (i.e. another commonly-used ANN model), lateral connections between RBFs (kernel units) are allowed; this manner of connection essentially removes the topological constraints in the network structure, whilst the MLP-NNs or RBF networks are based upon a strict layered structure. Moreover, such a manner also allows the connection between the kernel units with different domain inputs, which is not generally considered within the traditional ANN context; for example, an adaptive associative memory system that simultaneously deals with both auditory and visual input pattern data can be developed by kernel memory in a single framework of learning (Hoya 2004; 2005). Thus, in the case of kernel memory, knowledge is not represented in a mere distributed form (as in MLP-NNs) but rather by the nodes themselves (i.e. where each node may exhibit a generalisation capability) and their associations, the principle of which also inherits the important property of traditional symbolic approaches.

iii) The number and internal parameters of kernel units (i.e. both the centre positions and widths in the case of RBFs), as well as the weighted connections in between, can be dynamically varied according to the input data given to the network during the learning phase. For example, if we apply a simple Hebbian-motivated rule (to be described next) to configure the kernel memory (Hoya 2004; 2005), the network is self-organised by adding new (or otherwise removing unnecessary) kernel units and then by establishing/updating (or otherwise removing) the link weights in between, where appropriate (rather than starting off with a totally fixed and fully-connected architecture as in MLP-NNs). Hence, a more life-like network representation is possible in terms of kernel memory.

The points i) and ii) above correspond to the Hume's two laws of associations – similarity (i.e. when RBFs are chosen as kernel units) and contiguity (i.e. modelled by means of the link weight connections between kernel units) (Pinker 2000). On the other hand, related principally to the case of RBF kernel units, a neuropsychological justification was made in the work of 3D-vision study (Poggio and Edelman 1990), in which an RBF can be regarded as a plausible model of receptive field. The concept is also affirmative in light of the neuro-physiological study by Hubel and Wiesel (i.e. in terms of macaque monkey's visual cortex, Hubel and Wiesel 1977).

3.2 The Hebbian-motivated learning rule

As aforementioned, the Hebbian-inspired learning rule can be used to self-organise the kernel memory: suppose that we assign each kernel unit as an RBF and want to self-organise and use the memory for a particular pattern recognition task. Initially, there is only a single kernel unit in the memory space, i.e. a kernel unit K_1 with its centre position (in a high dimensional space) identical to the first pattern in the training set (and the auxiliary buffer attached to it to the corresponding class ID). In presenting the second training pattern, if K_1 does not activate (i.e. the value given by the similarity measurement between the first and second pattern exceeds a certain threshold), then add a new kernel K_2 with its centre position to the second pattern (and auxiliary buffer set to the corresponding ID). Then, for the third training pattern, the following three cases are considered; i.e. i) K_1 and K_2 are simultaneously activated, ii) either K_1 or K_2 is activated, iii) neither K_1 nor K_2 is activated. For each case, the following rules are respectively applied:

i) Establish a new link weight between K_1 and K_2. At later presentations, if this occurs repetitively, increment the weighting value with a small amount.
ii) Do nothing (since the pattern space is already covered).
iii) Add a new kernel unit K_3, with both the centre position and auxiliary buffer identical to those corresponding to the third pattern.

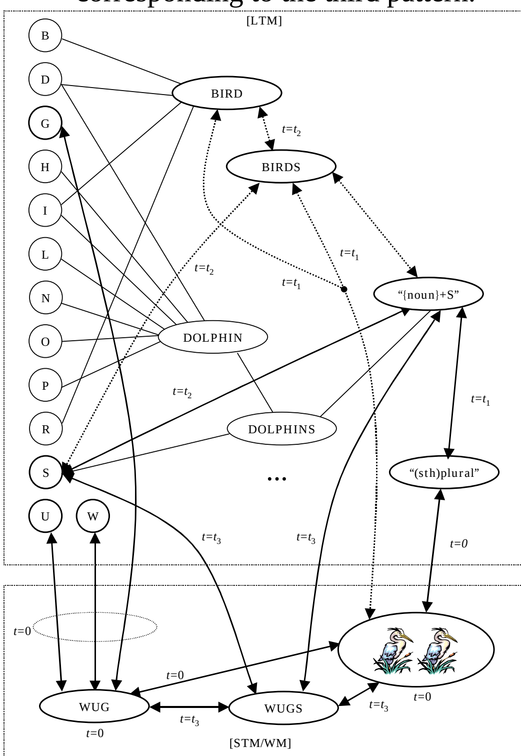

Fig. 1: Representation of the plural form WUGS by means of kernel memory

The manner of adding new kernel units and establishing the link weights continues until all the training patterns are presented to the kernel memory. In contrast, we may also introduce a shrinking rule – if the simultaneous (or subsequent) activation of K_i and K_j does not occur during a certain period of time, decrement the link weight value in between. Moreover, if there are some kernel units that have not activated for a longer period, we may remove not only the link weights but also such kernel units from the memory space.

It should be noted that use of the Hebbian-motivated rule described above does not involve any arduous and iterative approximation procedure (as in the back-propagation algorithm) at all and thereby that simple incremental learning is possible.

With the fundamental tenets given so far, we will next show that AMS with kernel memory can yield a reasonable basis to explain the faculty of 'artificial' language acquisition in terms of the solution to the Berko's "wug-test" (Berko 1958).

4 Solution to the Berko's "wug-test"

Pinker suggested that, for a similar problem to the wug-test, that is, inflection of verbs, the acquisition system must implement a hybrid of memory association (i.e. for irregular verbs) and rule-based mechanism (i.e. for regular verbs). Note that the wug-test only shows the visual result, and we will focus here on a visual sensory modality only. (Within the AMS context, a similar scenario for the auditory case is possible and considered to be relatively straightforward. Nevertheless, we are planning to propose it elsewhere in near future.)

4.1 Prerequisite

Suppose that i) each node in Fig. 1 is equivalent to a *population* of kernel units (with interconnections in between, where appropriate; or, a sub-kernel memory self-organised by e.g. the aforementioned Hebbian-motivated rule) representing a single stem of a particular data domain (i.e. a letter, word, or other concept) and that ii) AMS has already formed the overall composite network structure as shown, within the LTM in the previous acquisition/learning phase (i.e. innate or not). For convenience, let us also assume here that only RBFs are considered as the kernel units in each population (or node in Fig. 1). Then, since, as described earlier, each kernel unit performs a local pattern matching within the corresponding data domain, a collection of such matching results then the subsequent operation (i.e. such as max op.) leads to the overall pattern recognition result of a particular domain. Next, we will show how AMS processes the spelling pattern of a letter/word.

4.2 Pattern recognition of a single letter

First, AMS performs (visual) acquisition of the spelling pattern of a word by the sensation module. Second, provided that AMS has successfully performed the image segmentation/feature extraction (i.e. within the signal processing/machine learning context) of an entire image, corresponding to the spelling pattern, into separate objects (i.e. obtained as the feature patterns, corresponding to the respective letters 'B', 'I', 'R', and 'D'), the feature patterns are subsequently given as the input to all the populations of kernel units representing visually the alphabetic characters (i.e. as shown in an array of nodes 'B' to 'W', on the far left side in Fig. 1). Then, for each letter pattern, it is considered that some of the kernel units within some populations can activate (i.e. if the similarity measure given by an RBF yields a value that exceeds a certain threshold). Third, if we apply a simple winner-takes-all strategy, i.e. the population with a maximum number of the activated kernel units automatically corresponds to the pattern recognition result of a single letter, and the node (i.e. representing the population) will eventually emit a spike-like pulse.

4.3 Pattern recognition of a word

By performing subsequently the pattern recognition of each letter that constitutes the spelling pattern of the word, e.g. BIRD, a firing pattern consisting of the spike trains something like:

'B': 1 0 0 0
'I': 0 1 0 0
'R': 0 0 1 0
'D': 0 0 0 1

can be obtained (within a certain period of short time); in each column above, '1' or '0'means that a spike was or wasn't emitted from the corresponding node, at a particular time instance. Then, pattern matching of a similar firing pattern (i.e. given as the input) with the pattern as in the above stored within the word-level kernel units may accordingly result in the activation of some kernel units in a certain population (corresponding to a word node as in Fig. 1) and, similar to the single letter case, eventually yield the result of pattern recognition result at word level. (Note that this manner of processing can also avoid the ambiguity in the ordering of the letters, or the anagram problem, as pointed out by Pinker.)

4.4 Elimination process of the link weights

For the formation of word-level nodes representing the plural forms such as the BIRDS and DOLPHINS nodes in Fig. 1, two ways of representing the firing patterns can be considered: e.g. the BIRDS node can be formed via the firing pattern representing the subsequent activations of either i) the letter nodes 'B', 'I', 'R', 'D', 'S' or ii) the word node 'BIRD' followed by the letter 'S' node.

In general, it is naturally considered that a living system prefers a parsimonious solution instead of complex ones (albeit regardless of its biological plausibility, in a strict sense, at neuronal/cell level), for the efficient data processing to adapt (ultimately) itself to the incessantly changing environment. Thus, the same principle could also be applied to the formation of both the letter- and word-level nodes within the LTM of AMS: for the pair of words BIRD and BIRDS or DOLPHIN and DOLPHINS, an elimination process of the link weights can start to occur, since the emission of a spike from the BIRD/DOLPHIN node is considered to be always followed by that of the BIRDS/DOLPHINS node. Therefore, in order to represent e.g. the BIRDS node, the augmented firing pattern (on the left hand side in the below)

'B':	1 0 0 0 0
'I':	0 1 0 0 0
'R':	0 0 1 0 0
'D':	0 0 0 1 0
'S':	0 0 0 0 1

BIRD:	1 0
'S':	0 1

should be redundant; it is more parsimonious and thus preferable to have instead the representation on the right hand side above, since the total number of spike trains is dramatically reduced from 25(=5x5) to 4(=2x2). Thus, as shown in Fig. 1, the nodes BIRDS/DOLPHINS can eventually represent the firing pattern similar to the above, whilst eliminating the number of link weights, as well as internal parameters of the corresponding kernel units (i.e. the internal data corresponding to the firing patterns stored within the kernel units). Such process can be achieved after a sufficient amount of the repetitive processing (similar) spelling patterns by AMS.

Although the elimination process of the link weights above is discussed from a rather computationally economical point of view, its implication is rather significant, especially when one considers the actual development of 'artificial' language acquisition system.

4.5 Establishment of the link weights between word-level kernel units

For the formation of the word kernel units, the same Hebbian-motivated rule as described in Sect. 3.2 can be applied, with the extra need for taking into account the time course of activation (e.g. if a certain firing pattern as in the above appears during a short time period of time, then add a new kernel unit to generalise the pattern within the memory space; cf. Hoya 2005). Thus, it is possible that, during consolidation of the LTM, the word nodes (i.e. BIRD, BIRDS, DOLPHIN, and DOLPHINS), as well as the cross-domain link weights between the corresponding letter- and word-level nodes (kernel units), are formed.

Moreover, it is considered that, with an appropriate pattern presentation setting, the link weights are correctly established between nouns and their plural forms and can be accordingly strengthened. For instance, the presentation of both the word BIRD and its plural form BIRDS within a short time interval can facilitate such an establishment, since this can lead to the simultaneous activations (or, more specifically, subsequent activations in a brief moment) amongst the corresponding kernel units. Then, the same notion applies to learn irregular forms, such as goose – geese.

4.6 Generalisation of the plurality – formation of super-ordinate nodes in the LTM

Besides the nodes representing letters/words, it is considered that both the super-ordinate nodes '{noun}+S' and 'something is plural' generalising the notion of plurality, as in Fig. 1, have also been formed during the consolidation of the LTM. As shown, the super-ordinate nodes can be activated, if AMS detects that the visual scene consists of multiple objects, and then functions to relay the activation to the kernel units connected, i.e. those representing the corresponding letters, spelling patterns of words, other modality-specific patterns (i.e. the visual images of birds, dolphins, etc), and/or notions, via the cross-domain link weights. In other words, such a node plays a similar role in representing a certain concept (as in ordinary pattern recognition, for example, that can correspond to the kernel unit representing a category label, within the kernel memory). Thereby, regular inflection of a noun into the corresponding plural form can be eventually represented by the activation of both the super-ordinate nodes, whilst only the super-ordinate node 'something is plural' can be activated in the case of irregular nouns; in the goose – geese case, only the subsequent activation of GOOSE -> GEESE -> 'something is plural' may occur by the memory access in LTM.

4.7 Processing of the unknown word WUG

Now, let us consider a situation where AMS acquired visually the spelling pattern of the unknown word WUG (arrived in the STM/WM, at time t=0) and successfully performed the subsequent pattern recognition of the respective letters ‘W’, ‘U’, and ‘G’ and where it also processed the visual image within the STM/WM and detected that the visual scene is composed by multiple (bird-like) objects (t=0).

Then, as described in the previous subsection, the super-ordinate node of ‘something is plural’ can be also activated (t=0), relay the activation to the other super-ordinate node ‘{noun}+S’, and it can eventually invoke the activation of the ‘S’ node. Thereby, such a situation is considered where the subsequent activation, i.e. the activation of the node representing the unknown word ‘WUG’ within the STM/WM followed by that of the letter node ‘S’, occurs. This is similar to the case of the formation of the BIRDS node from the BIRD node (in Sect. 4.4). Therefore, a new node representing the plural form WUGS, as well as the link weights between the corresponding kernel units ($t=t_3$), is temporally created within the STM/WM module, which suggests that repetitive presentation of such a pattern consolidates the firing pattern and eventually makes both the WUG and WUGS nodes (as well as the link weights) a part of the LTM within AMS.

5 Conclusion

In this paper, we have proposed a connectionist model that accounts for a solution to the Berko’s wug-test in terms of the pattern recognition process via the STM/WM and LTM within AMS. Although the study has focused upon a visual aspect, we believe that the extension to the auditory case is relatively straightforward (albeit some additional requisite specific to auditory data processing, e.g. to describe why the native speakers tend to pronounce WUGS /-ɪz/, neither /-s/ nor /-z/), which has been crucial within general linguistics context and is therefore currently under investigation. (In a similar context, the inflection of verbs can be also explained.) It should however be emphasised that, within our approach, pattern recognition at both the letter- and word levels can be performed (Sects. 4.3 - 4.5) within a single framework of the Hebbian-motivated learning, in parallel to generalisation of the inflections (Sects. 4.6 and 4.7), which is not generally considered in conventional connectionist accounts. Moreover, the proposed single framework also agrees with the “memory association then comes the rule” principle suggested by Pinker (Pinker 2000). This is implicitly depicted in Fig. 1 – the memory association is performed faster than the rule induction (i.e. the word pattern arrived at the STM/WM is processed faster by direct access to the nodes in LTM, rather than via the subsequent relay by the super-ordinate nodes ‘something is plural’ and ‘{noun}+S’). Future work also includes performing an actual simulation study using computers to confirm our proposal.

References

Berko, J. 1958. The child’s learning of English morphology. *Word, 14*, 150-177.

Fodor, J. 1983. *The Modularity of Mind*. MIT Press.

Hobson, J. 1999. *Brain and Consciousness*.

Hoya, T. 2004. Self-organising associative memory for multi-domain pattern classification, *Proc. ALCOSP-04*, 735-740.

Hoya, T. 2005. *Artificial Mind System – Kernel Memory Approach*. Springer-Verlag:Heidelberg.

Hubel, D. H. and Wiesel, T. N. 1977. The Ferrier lecture: functional architecture of macaque monkey visual cortex. *Proc. of R. Acad. Lond., Series B 198*, 1-59.

Pinker, S. 2000. *Words and Rules – The Ingredients of Language*. Perennial.

Poggio, T. and Edelman, S, 1990. A network that learns to recognize three-dimensional objects. *Nature, 343-18,* 263-266.

Rumelhart, D. E., McClleland, J. L., and the PDP Research Group. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1. Foundations*, MIT Press: MA, Cambridge.

Zalta, E. N. (ed). 2006. *Stanford Encyclopedia of Philosophy – Connectionism* (on line at: http://plato.stanford.edu).