MINIMAL PREREQUISITS FOR PROCESSING LANGUAGE STRUCTURE: A MODEL BASED ON CHUNKING AND SEQUENCE MEMORY

ANNA JON-AND[*1,2] and JÉRÔME MICHAUD[3,4]

[*]Corresponding Author: anna.jon-and@su.se
[1]Centre for Cultural Evolution, Stockholm University, Stockholm, Sweden
[2]School of Humanities and Media Studies, Dalarna University, Falun, Sweden
[3]Department of Sociology, University of Uppsala, Uppsala, Sweden
[4]Department of Physics and Astronomy, University of Uppsala, Uppsala, Sweden

In this paper, we address the question of what minimal cognitive features are necessary for learning to process and extract grammatical structure from language. We build a minimalistic computational model containing only the two core features *chunking* and *sequence memory* and test its capacity to identify sentence borders and parse sentences in two artificial languages. The model has no prior linguistic knowledge and learns only by reinforcement of the identification of meaningful units. In simulations, the model turns out to be successful at its tasks, indicating that it is a good starting point for an extended model with ability to process and extract grammatical structure from larger corpora of natural language. We conclude that a model with the features chunking and sequence memory, that should in the future be complemented with the ability to establish hierarchical schemas, has the potential of describing the emergence of grammatical categories through language learning.

1. Introduction

In the search for the cognitive mechanisms underlying human language learning capacity, chunking has been identified as essential for overcoming memory constraints in online language processing (Christiansen & Chater 2016). The idea that language acquisition occurs through language use and through continuous updating of linguistic knowledge encoded as chunks or constructions (Bybee 1985, Tomasello 2003) has recently been successfully implemented in a chunk-based language acquisition model (McCauley & Christiansen 2019).

Sequence learning has also been pointed out as central for the language capacity (Bybee 2002, Christiansen et al. 2002, Frank et al. 2012). Faithful sequence representation has additionally been suggested to be uniquely human (Grant & Roberts 1976, MacDonald 1993, Roberts 2002, Ghirlanda et al 2017). Combined with chunking, sequence memory enables the storing of the sequential order of a chunk's components. The combination of these two features allows for the successive building up of a hierarchy of chunks that can support the identification of meaningful constructions in language processing.

The aim of this paper is to test whether the two features chunking and sequence memory are sufficient to extract simple grammatical structure from strings of artificial languages containing structures that are typical for natural languages. By implementing a minimal model architecture where no linguistic properties are predefined and evaluate its ability to extract sentences and grammatical structure in simple artificial languages, we aim at commenting the potential of the model to represent core features of the human language learning capacity. Furthermore, we compare a model with hierarchical chunking capacity to a model with a simpler incremental chunking capacity, in order to discuss whether and how chunking supports language learning. We also aim at discussing whether this model is a good starting point for an extended model able to process and extract grammatical structure from larger corpora of natural language.

2. Model

The task of the model is to segment a stream of incoming stimuli into meaningful units, conceptualized as sentences. The input consists in two small artificial languages with simple grammars. The first language contains one transitive verb and two nouns that can have the syntactic functions of subjects or objects. The word order of the language can be subject-verb-object or object-verb-subject, depending on how it is parsed. The first language thus consists of four sentences:

(1) noun_1 verb noun_1
(2) noun_1 verb noun_2
(3) noun_2 verb noun_1
(4) noun_2 verb noun_2

The second language is similar to the first one but increases complexity by introducing the possibility of adding a subordinate clause after each noun. The subordinate clause consists of a verb and a noun, making the language recursive. For both language conditions, sentences are repeated randomly in a string that constitutes the input to be processed by the model's learning mechanism. The input contains no cues that reveal the sentence borders. The processing of the input is performed by an associative learning mechanism in which $v(s \rightarrow b)$ is the stimulus-response association between stimulus *s* and behaviour *b* which estimates the value *v* of performing behaviour *b* when encountering stimulus *s*. As a consequence of experience, an agent learns about the value of responding with *b* to *s* according to

$$\Delta v(s \rightarrow b) = \alpha [u - v(s \rightarrow b)] ,$$

where *u* is the reinforcement value and α regulates the rate of learning. A target behaviour *b* is associated with every stimulus and the corresponding reinforcement value *u* is positive. Otherwise, the reinforcement value *u* is negative.

In our learning simulations, the learning mechanism perceives two elements and their internal temporal order before each decision. The first element can be atomic or complex, depending on previous chunkings. When perceiving a sequence, the mechanism has two basic possibilities of behaviours:

(i) ***Place border*:** A border is placed between the first and the second element in the pair. The model then suggests that a sentence ends where the border is placed and begins where the last border was placed.

(ii) ***Chunk*:** The two elements will then form a chunk that will constitute the first element in the next perceived sequence. If the first element is already a chunk, different kinds of chunkings may occur.:

   a. ***Right-chunk*:** The last element is chunked on the right-hand side to the first, without changing the internal structure of the first element.
   b. ***Sub-chunk:*** The last element is chunked with a sub-element in the first element, causing a restructuring of the first element. The number of sub-elements to which the last element can be chunked is determined by the structure of the first element. In a binary tree structure, chunking can occur with any element or node that is accessible from the right-hand side of the tree, illustrated in Fig. 1.

Flexible chunking (where right-chunking or any accessible sub-chunking can occur) generates binary hierarchical tree structures that can have any number of left- or right-branches. Fig. 1 illustrates possible chunkings with a complex element. In the example, the last element has four chunking possibilities. The upper cross indicates right-chunking and the three subsequent crosses indicate chunkings at increasingly lower levels.

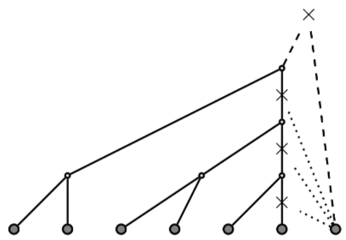

Figure 1. Illustration of possible chunkings when the first element is complex.

In order to investigate whether flexible chunking supports learning, we also test a simpler mechanism that can only right-chunk, generating purely left-branching trees. We call the two mechanisms ***flexible chunking*** and ***right-chunking***.

If a border is placed, positive reinforcement is given for correct identification of sentence boundaries and negative reinforcement for identification of incorrect boundaries. Reinforcement implies strengthening or weakening the association between the perceived sequence and the performed behaviour. This represents the concept that a language learner receives an internal or external reward for the identification of a meaningful unit i.e. a sentence. Trying to make sense of a nonsensical unit, on the other hand, can generate frustration, represented by negative reinforcement in the model. When reinforcement is given, it is also back-propagated to preceding chunkings that contributed to the successful or unsuccessful sentence identification.

Like a naïve language learner, the mechanisms in the model have no prior knowledge of grammatical structure. The mechanisms need to explore and discover on their own the chunkings that lead to correct sentence identifications and positive reinforcement.

3. Results

Results from simulations show that both the flexible chunking mechanism and the right-chunking mechanism learn to identify the sentences in the input in the two languages. As can be seen in Fig. 2, all sentences in the first, less complex language are identified and consequently pointed out after a learning process that takes the shape of an S-curve. The learning curves for the two mechanisms are very similar with the only difference that the curve of the *flexible chunking* *mechanism* has a slightly steeper S-shape. This is likely due to the larger behaviour repertoire resulting from flexible chunking, that slows down learning initially. Once succesful chunkings are identified, learning is likely faster because both productive right-chunkings and sub-chunkings are reinforced. It is not clear, however, which of the two mechanisms learns fastest.

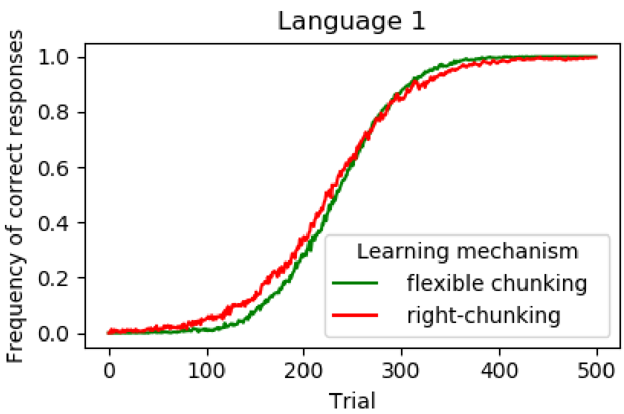

Figure 2. Simple sentences. Learning curves of the two mechanism based on sub-chunking and incremental chunking. The curves averages correct responses over 250 simulations.

After a complete learning process, both mechanisms had extracted all four grammatical sentences in the language and no agrammatical sentences. An example of grammars and parsings extracted by the mechanisms is presented in Table 1. As can be seen in Table 1, flexible chunking generates both left-branching and right-branching parsings. There seems to be no tendency for right-branching or left-branching; they are equally favoured. This is likely due to the fact that a chunking followed by a chunking or a sub-chunking are equally likely to occur and both lead to successful border placement. Once one of the two variants is tried out, it is reinforced, and the mechanism sticks to it. The right-chunking mechanism, on the other hand, generates only left-branching parsings.

Table 1. Grammars for the simple first language extracted by the two learning mechanisms

	FLEXIBLE CHUNKING	***RIGHT-CHUNKING***
Correct sentence	*Parsing*	*Parsing*
$noun_2$ verb $noun_1$	(($noun_2$ verb) $noun_1$)	(($noun_2$ verb) $noun_1$)
$noun_2$ verb $noun_2$	(($noun_2$ verb) $noun_2$)	(($noun_2$ verb) $noun_2$)
$noun_1$ verb $noun_2$	($noun_1$ (verb $noun_2$))	(($noun_1$ verb) $noun_2$)
$noun_1$ verb $noun_1$	($noun_1$ (verb $noun_1$))	(($noun_1$ verb) $noun_1$)

The second and more complex language, that involves subordinate clauses, generates a more interesting result. As seen in Fig. 3, learning to correctly identify sentences in the more complex language takes much longer for the two learning mechanisms, but it is now clear that the flexible chunking mechanism learns much faster than the right-chunking mechanism An analysis of the parsing generated by the flexible chunking mechanism offers a possible explanation to this.

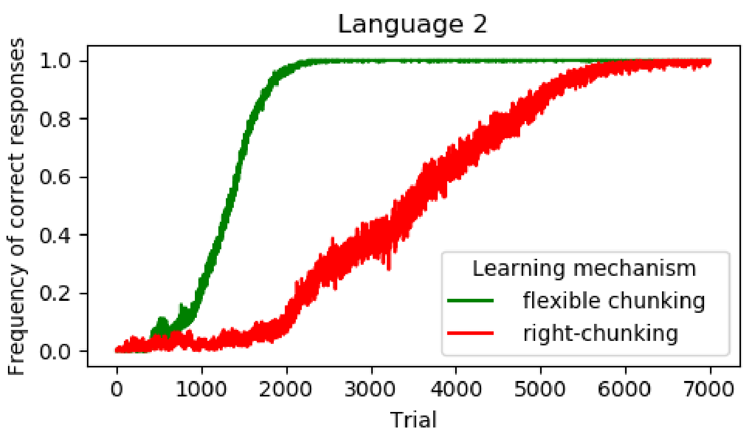

Figure 3. More complex sentences with subordinate clauses. The probability of a subordinate clause appearing after a noun is here 0.5 and the maximum number of subordinate clauses per sentence is 2. Learning curves of the two mechanisms based on flexible chunking and right-chunking. The curves average correct responses over 250 simulations.

In Table 2, only parsings generated by flexible chunking are presented. Right-chunking consistently generates left-branching parsings that do not need to be illustrated once more. As the number of possible sentences is high, some example parsings are demonstrated to illustrate the tendency that was identified.

Table 2. Examples from a grammar extracted by the flexible chunking mechanism in the more complex language with subordinate clauses.

	FLEXIBLE CHUNKING
Correct sentence	*Parsing*
noun_2 verb noun_1	((noun_2 verb) noun_1)
noun_1 verb noun_2	(noun_1 (verb noun_2))
noun_1 verb noun_2 verb noun_2	((noun_1 verb) (noun_2 (verb noun_2)))
noun_1 verb noun_1 verb noun_1 verb noun_2	(noun_1 (verb ((noun_1 verb) (noun_1 (verb noun_2)))))

As can be seen in Table 2, there is still no consequent right-branching or left-branching parsing of the recursive sentence structures with subordinate clauses. What can be observed, however, is how chunkings from shorter sentences support the building up of longer sentences.

Firstly, a chunking that is always reinforced in the identification of the shorter sentences is noun-verb. This supports the sub-chunking of a noun and a verb in subordinate clauses. In the parsing of sentences with subordinate clauses, nouns followed by a verb are always preceded by a left parenthesis, indicating that the noun and the following verb have been sub-chunked. Secondly, if a sub-chunking of the verb and a noun occurs in the parsing of a short sentence, as in the second example sentence in Table 2, where the verb is sub-chunked with noun_2, this sub-chunking tends to reappear in subordinate clauses. This can be seen in the last two sentences in Table 2, where this sub-chunk appears last in the parsing of both sentences.

Apart from these two tendencies, different and seemingly random parsing structures appear. It seems clear, however, that frequent chunkings in shorter sentences are reused in longer sentences. This probably explains the fast learning of the flexible chunking mechanism. The right-chunking mechanism cannot use the support from previous chunkings when learning to identify increasingly longer sentences.

4. Discussion

These first results from testing the language processing capacity of a minimal language learning model are promising for future extensions of the model. The fact that a reinforcement learning model including only the two core features *chunking* and *sequence memory* is able to learn to correctly identify sentences in small artificial languages with and without recursion is a preliminary yet powerful indication of the potential of the model. The ability of a model to extract meaningful constructions with no pre-assumptions concerning grammatical categories or rules is compatible with the idea of emergent grammatical categories in Radical Construction Grammar (Croft 2001). The comparison between the flexible chunking mechanism and the right-chunking mechanism shows that flexible chunking initially slows learning down, but if the language is complex and contains repetition of structures, learning soon becomes faster than for the right-chunking mechanism. This indicates that flexible chunking may be an important property of an incremental learning model based on chunking and sequence memory. We believe that if exposed to more complex and variable grammatical structures, such as those of natural languages, and extended with a schematizing feature, the model would most likely chose right-branching or left-branching parsing of given structures more consistently and this would likely increase the utility of flexible chunking even more.

The principle of flexible chunking is similar to that of *unsupervised data-oriented parsing* (U-DOP) (Bod 2006), in the sense that U-DOP can generate any tree-structure with no lexical or structural constraints. However, while U-DOP requires costly computations to estimate the most probable parse trees, our model provides the same flexibility implemented in a simpler way.

A feature that we believe should be added to the model in the future is the ability to establish schemas based on the similarity of the strings and structures that enter the decision function. This feature should reduce learning costs, which will be necessary for processing natural language corpora with a much higher diversity and complexity than the small artificial languages used here. The schematizing feature may also generate increasingly abstract schemas organized in a hierarchical network that can be studied and compared with conventional grammatical descriptions of a language. A possible future application of the model is thus to describe the emergence of lexico-grammatical categories through language learning,

References

Bod, R. (2006, July). An all-subtrees approach to unsupervised parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 865-872). Association for Computational Linguistics.

Bybee, J. (1985). Morphology: A Study of the Relation between Meaning and Form. Amsterdam: John Benjamins.

Bybee, J. (2002). Sequentiality as the basis of constituent structure. Typological Studies in Language, 53, 109-134.

Croft, W. (2001). Radical construction grammar: Syntactic theory in typological perspective. Oxford University Press on Demand.

Christiansen, M. H., Dale, R. A., Ellefson, M. R., & Conway, C. M. (2002). The role of sequential learning in language evolution: Computational and experimental studies. In Simulating the evolution of language (pp. 165-187). Springer, London.

Christiansen, M. H., & Chater, N. (2016). The Now-or-Never bottleneck: A fundamental constraint on language. Behavioral and Brain Sciences, 39.

Frank, S. L., Bod, R., & Christiansen, M. H. (2012). How hierarchical is language use?. Proc. R. Soc. B, 279(1747), 4522-4531.

Ghirlanda, S., Lind, J., & Enquist, M. (2017). Memory for stimulus sequences: a divide between humans and other animals?. Royal Society open science, 4(6), 161011.

Grant, D. S., & Roberts, W. A. (1976). Sources of retroactive inhibition in pigeon short-term memory. Journal of Experimental Psychology: Animal Behavior Processes, 2(1), 1.

McCauley, S. M., & Christiansen, M. H. (2019). Language learning as language use: A cross-linguistic model of child language development. Psychological review, 126(1), 1.

MacDonald, S. E. (1993). Delayed matching-to-successive-samples in pigeons: Short-term memory for item and order information. Animal Learning & Behavior, 21(1), 59-67.

Roberts, W. A. (2002). Are animals stuck in time?. *Psychological bulletin*, *128*(3), 473.

Tomasello, M. (2003). *Constructing a language: A usage-based theory of language acquisition.* Cambridge, MA: Harvard University Press.