Usage-based Grammar Induction from Minimal Cognitive Principles

Anna Jon-And
Stockholm University, Sweden
Centre for Cultural Evolution
Department of Psychology
Department of Romance Studies
and Classics
anna.jon-and@su.se

Jérôme Michaud
Stockholm University, Sweden
Mälardalen University
Centre for Cultural Evolution
Department of Psychology
Västerås, Sweden
jerome.michaud@mdu.se

This study explores the cognitive mechanisms underlying human language acquisition through grammar induction by a minimal cognitive architecture, with a short and flexible sequence memory as its most central feature. We use reinforcement learning for the task of identifying sentences in a stream of words from artificial languages. Results demonstrate the model's ability to identify frequent and informative multi-word chunks, reproducing characteristics of natural language acquisition. The model successfully navigates varying degrees of linguistic complexity, exposing efficient adaptation to combinatorial challenges through the reuse of sequential patterns. The emergence of parsimonious tree structures suggests an optimization for the sentence identification task, balancing economy and information. The cognitive architecture reflects aspects of human memory systems and decision-making processes, enhancing its cognitive plausibility. While the model exhibits limitations in generalization and semantic representation, its minimalist nature offers insights into some fundamental mechanisms of language learning. Our study demonstrates the power of this simple architecture and stresses the importance of sequence memory in language learning. Since other animals do not seem to have faithful sequence memory, this may be a key to understanding why only humans have developed complex languages.

Action Editors: Marianna Apidianaki, Abdellah Fourtassi, and Sebastian Padó. Submission received: 17 December 2023; revised version received: 30 April 2024; accepted for publication: 19 June 2024.

https://doi.org/10.1162/coli_a_00528

1. Introduction

Human language capacity stands out among species for its flexibility and expressive power. A fundamental unsolved question is what cognitive mechanisms enable humans to learn language. On the one side, there are theories on inborn language organization (Chomsky 1957; Pinker and Jackendoff 2005; Fodor 1983) or language specific learning processes (Nowak, Komarova, and Niyogi 2002; Reali and Griffiths 2009; Griffiths et al. 2010; Tenenbaum et al. 2011). On the other side, we find theories emphasizing usage-based, general learning principles (Bybee 1985; Tomasello 2003; Heyes 2018), combined with ideas on culturally emergent structure (Kirby, Cornish, and Smith 2008; Goldberg 2007; Langacker 1987; Croft 2001). Explanations relying on strong functional modularity often fail to explain variability across languages and disregard that human mental skills require social learning. Moreover, recent advancements in large language models (LLMs) challenge the notion that predetermined linguistic predispositions are necessary for language learning (Piantadosi 2023). In contrast, domain-general learning theories often lack explicit suggestions on the machinery that would enable such learning and leave the question of why other animals cannot acquire human-like language unanswered. In this article, our point of departure is the novel **sequence hypothesis**, postulating that faithful sequence representation is uniquely human and a central prerequisite for the emergence of language and complex culture (Ghirlanda, Lind, and Enquist 2017; Lind, Ghirlanda, and Enquist 2022; Enquist, Ghirlanda, and Lind 2023; Lind et al. 2023; Jon-And et al. 2023). Based on this, we propose a minimal cognitive architecture with a short and flexible sequence memory as its most central component and examine how far this design can reach in learning grammar from data.

Our proposed cognitive architecture follows a simple and traceable error-correction temporal difference learning algorithm (Sutton and Barto 2018). In order to make the model fully interpretable, we use an equation-based model rather than a neural network. Our minimal architecture allows us, on the one hand, to discuss what cognitive properties are necessary for language learning and, on the other hand, to follow the process of extracting grammatical information that supports the language learning process. Our approach is inspired by cognitive architecture research in artificial intelligence, that aims at building a unified theory of cognition where few general mechanisms should be able to explain diverse phenomena and characteristics of human cognition (Newell 1994). In line with recent domain-general artificial intelligence modeling (Wiggins 2020; van der Velde et al. 2017), we aim at accounting for human information processing while making as few assumptions as possible. Our initial hypothesis, drawing upon a pilot study for the present work (Jon-And and Michaud 2020), is that sequence memory, paired with chunking, are key domain-general cognitive mechanisms underlying the human ability to learn grammar from data. Our approach implements principles of usage-based language learning, namely, that language acquisition occurs through language use and through continuous updating of linguistic knowledge encoded as chunks or constructions (Bybee 2006; Tomasello 2003).

We explore reinforcement learning as a foundational framework for our learning model, seeking a cognitively plausible architecture and learning process, that can generate language structure. Reinforcement learning is grounded in robust empirical support for organisms learning through trial-and-error interactions with their environment (Pavlov 1949; Skinner 1965; Mackintosh 1983; Rescorla and Wagner 1972; Sutton and Barto 2018). Our objective is also to use simple, localized learning principles to model the acquisition of a complex system like grammar. The use of reinforcement learning aligns well with this aim, as reinforcement learning is suggested to be a sufficient mechanism to acquire many sophisticated and multifaceted behaviors that are found in humans, animals, and machines (Silver et al. 2021).

The task of our learning model is to identify sentences in a linguistic stream where cues to sentence borders like punctuation or capitalization are removed. In our model, the learner follows a chunk-and-pass learning principle (Christiansen and Chater 2016b) in which it decides whether to place a boundary or to chunk a newly encountered word to other words kept in the working memory. This is comparable to the shift-and-reduce procedure applied by Yogatama et al. (2016), where tree structures emerge and are optimized to improve performance on more advanced linguistic tasks like sentiment analysis and semantic judgments. In Yogatama et al. (2016), this is done by using word embeddings that emerge in a neural network, making both the tasks and the model considerably more complex than ours. In contrast, segmentation is a task that is present in early language learning, when structure is likely established (Peters 2013). Our chunking mechanism creates hierarchies with binary tree structures with the words as leaves. Since chunks retain the order in which words are encountered, they encode both the ability for sequence representation and chunking. Boundary placement triggers reinforcement, where correct sentence identification is rewarded and incorrect sentence identification is penalized. When referring to reward or penalization during language learning, we do not assume explicit feedback, that is often absent in language learning. Our assumption, rather, is that ultimately, successful communication triggers internal or external reinforcement, being the main goal of language use. Language understanding is a central part of successful communication, and the segmentation of a linguistic stream into meaningful units is in its turn necessary for understanding. While studies on segmentation in language learning have focused on the word level (Saffran, Aslin, and Newport 1996; Saffran 2001), higher levels of segmentation are also necessary for language understanding. Segmentation tasks are arguably more central for language learning than, for example, word or sound prediction tasks.

We choose the task of segmenting a stream of words into sentences because we are interested in grammar induction. We hypothesize that in a minimalist learning model that makes no assumptions on specific genetic guidance for language learning, and imposes cognitively plausible constraints on working memory and processing capacities, grammar emerges as a solution to the combinatorial challenge posed by language learning. The expectation is that hierarchical organization of words will emerge as a support for the task of sentence identification. There are, naturally, other tasks in language processing and production that are supported by grammar and may trigger its emergence. In this study, we have chosen one easily encoded and evaluated task, and test if this task is a sufficient driving force for grammar induction. This task may in future studies be combined with or compared to other tasks. We focus on syntax even though syntax and morphology are closely intertwined, for reasons of simplicity and for facilitating future testing of the model on written natural language corpora. Furthermore, while sentence boundaries are nowhere near as clear and consequent in spoken spontaneous language as in written texts, the identification of multi-word meaningful units is always a relevant task in language understanding. Here, for reasons of simplicity, we model sentences as a meaningful unit to be identified, to enable evaluation of the fundamental principles of our model.

Our model does not formally encode meaning, even though meaning is central to the concept of a linguistic construction, that is essentially a pairing of form and meaning (Langacker 1987; Goldberg 2007; Croft and Cruse 2004; Tomasello 2005). The notion of meaning is not, however, entirely absent from our model, as an underlying assumption is that identification of meaningful units is a prerequisite for a form-meaning mapping to occur. But the support for segmentation provided by meaning in natural language learning, as well as prosodic cues like stress or pauses (Kuhl 2004; Sanders, Neville, and Woldorff 2002) are absent in our model. These simplifications may impose difficulties for our learner to succeed in its task when linguistic complexity increases, but they also enable investigating how far the system can reach without this additional support.

Our work bears similarities to unsupervised grammar induction, which involves inferring a grammar from strings or sentences within a language. However, differently from our architecture, most unsupervised grammar induction models assume the existence of given formal syntactic relations even if there is no explicit training on how these should be encoded (Muralidaran, Spasić, and Knight 2021). While exceptions to this top–down perspective exist (Bod 2006; Solan et al. 2003; Shain et al. 2016), our goals also depart from those of unsupervised grammar induction in the sense that we are not searching for the model that best predict data. Instead, we are aiming at modeling the acquisition of grammar in a minimalist and cognitively plausible manner that is consistent with usage-based learning, and at testing whether sequence memory is sufficient to achieve this. In syntactic parsing the explicit task is to identify trees, dependencies, and/or constituents in sentences, while the task of our learner is to identify sentence boundaries, a goal more directly related to language understanding, and trees emerge as a support for this task.

The evaluation of the model is based on the one hand on its ability to fully learn to identify sentences in artificial languages with varying degrees of complexity, and on the other hand on its ability to identify and reuse chunks that are extracted from exposure to these languages. The second more qualitative measure is more central than the model's quantitative performance, as our goal at this stage is primarily to study whether and how our design allows for extraction of functional structure. In order to understand how structure emerges during learning in our model, and to discuss whether this process in any aspects resembles natural language learning, we study the learning process and not only the final result. The evaluation is performed on relatively small artificial languages reflecting common natural language structures. To scale to larger input, the model needs a system that allows for abstraction, and currently it only contains a first step towards generalization, as the reuse of frequent chunks, partially reduces the combinatorial explosion of language learning. This partial solution can serve as a basis for generalization that may give rise to grammatical abstractions at higher levels. Such generalization could be implemented in, for example, a neural network, but if the transparency of the model is to be maintained, generalizations can be implemented within the error-correction temporal difference learning algorithm applied here. One way of doing this is using a category-based formalism that makes no pre-assumptions on categories, but only generalizes over the properties of the useful chunks the learner identifies. A pilot implementation of such a generalization system has been conducted by Jon-And and Michaud (2024).

As the system presented here does not include large-scale generalization, we do not evaluate the model's performance on natural languages. In artificial languages, we find that the model is able to efficiently identify informative chunks and reuse them when they occur in new structures, thus in a simplified manner accounting for the emergence of grammar during learning. We also find that the model over time learns to ignore less informative chunks and gives priority to those that contribute more frequently to the correct identification of sentences. In this way, we see how a simple goal like identification of sentence borders gives rise to self-organization and consolidation in an emergent grammatical system.

This article is organized as follows: Section 2 provides some background for this study. Section 3 presents the computational framework used, that is, the framework of Markov Decision Processes (Sutton and Barto 2018). In particular, we discuss the learning task as well as the learning algorithms studied in this article. Section 4 presents the results of our study and discusses the performance of our model on a number of small human-like toy languages. We show that our model successfully learns these languages and provide some interesting insights into how learning is performed. Finally, Section 5 provides a discussion of the implication of our results and outlines how this work could be continued.

2. Background

2.1 Foundation of the Human Linguistic Capacity: Associative Learning, Sequence Memory, and Chunking

Associative learning is a fundamental mechanism that holds strong explanatory power for general learning in both humans and other animals (Pavlov 1949; Skinner 1965; Mackintosh 1983; Rescorla and Wagner 1972; Heyes 2018; Wasserman, Kain, and O'Donoghue 2023; Lind 2018; Heyes 2012b, 2012a; Enquist, Lind, and Ghirlanda 2016; Bouton 2016; Haselgrove 2016; Enquist, Ghirlanda, and Lind 2023). However, the inability of non-human animals to match the language capacities of humans calls for identifying unique properties that are present in humans, and not in other animals. Sequence learning has been pointed out as particularly important for human linguistic capacities (Heyes 2018; Christiansen and MacDonald 2009; Bybee 2002a; Christiansen and Kirby 2003; Christiansen and Arnon 2017; Frank, Bod, and Christiansen 2012; Cornish et al. 2017; Udden et al. 2012; Kolodny, Edelman, and Lotem 2015; Kolodny and Edelman 2018). At the same time, there is strong empirical support for non-human animals' limited capacity to represent the exact order of stimuli (Roberts and Grant 1976; MacDonald 1993; Ghirlanda, Lind, and Enquist 2017; Read, Manrique, and Walker 2022; Lind et al. 2023). This suggests faithful sequence representation to be a basic defining feature of human cognition and linguistic abilities (Enquist, Ghirlanda, and Lind 2023; Jon-And et al. 2023). These findings call for testing the capacity for language learning of a minimalist associative learning model with a precise but limited sequence memory.

To enable processing of language or of any kind of sequential information, sequence memory likely needs to be combined with a capacity for chunking, that is, considering a recurrent sequence of stimuli of flexible length as a unit. Without chunking, it is impossible to faithfully represent a sequence. Chunking is known to be central for human language learning (Tomasello 2003; Bybee 2002a; Servan-Schreiber and Anderson 1990; McCauley and Christiansen 2019). Humans have strong memory constraints in online language processing and are, for example, typically only able to store somewhere between three and ten meaningful items in their working memory (Cowan 2001; Miller 1956). The chunk-and-pass principle has been identified as essential for overcoming such memory constraints (Christiansen and Chater 2016b), and is also theorized to underpin the multilevel representational structure of language (Christiansen and Chater 2016a).

Principles similar to associative learning, of local decision-making and memory updating, rooted in the utilization of limited recent information and associations derived from past experiences, have been implemented in models like the naming game, where categorizations self-organize driven by communicative success and alignment (Steels and Loetzsch 2012), or the utterance selection model, that accounts well for several aspects of the change and diffusion of linguistic forms (Baxter et al. 2006; Michaud 2019). In these models, only knowledge pertaining to the features present in each specific learning occasion is updated, thus avoiding costly computations. These models focus on single linguistic elements and not sequences, and cannot, therefore, infer any system that encodes relationships between elements, like grammar. We hypothesize that by applying similar principles to sequential linguistic input, a global system of rules may emerge during learning and use without the learner having to possess explicit knowledge or hypothesize about the entire system. Instead, agents will rely upon simple emergent dynamic rules to inform each decision locally.

2.2 Connectionist Language Learning Models

In connectionist models, associative learning is operationalized through the adjustment of connection weights between artificial neurons. Such models, implemented in artificial neural networks, mimic the interconnected structure of neurons in the brain and tend to exhibit high performance in generalization of patterns from data. While connectionist models have successfully accounted for certain aspects of language acquisition (Elman 1990, 1996; Christiansen and Chater 2001; McClelland et al. 2010), their interpretability remains a challenge, and they struggle to capture symbolic representations.

Modern LLMs are connectionist models that stand out for their remarkable and often human-like linguistic performance. It is possible to gain some insight into what kind of information LLMs are able to store at the semantic, syntactic, and morphological levels by studying their attentional mechanisms (Manning et al. 2020; Vaswani et al. 2017; Piantadosi 2023; Piantasodi and Hill 2022), but the neural networks upon which LLMs rely contain billions to trillions of parameters (Piantadosi 2023; Brown et al. 2020; Amaratunga 2023) and it is not possible to extract precise information on whether or how any morpho-syntactic abstractions emerge in these models and, if so, what is their nature. In terms of cognitive plausiblity, there are also some divergences between the learning processes of LLMs and humans. LLMs typically leverage datasets approximately 1,000 times larger than the linguistic input available to a child (Warstadt and Bowman 2022). Moreover, humans learn and use language in parallel (Bybee 2006; Ellis, O'Donnell, and Römer 2015; Tomasello 2003), while LLMs are generally pre-trained and no longer learn when they are put in use. Both during training and use, LLMs can base their output on very long input sequences, sometimes containing thousands of tokens (Beltagy, Peters, and Cohan 2020; Liang et al. 2023; Brown et al. 2020), while human working memory is constrained to storing the aforementioned three to ten items (Cowan 2001; Miller 1956). Overall, LLMs have access to more resources in terms of both data, memory and processing power than humans during language learning. We hypothesize that imposing working memory and processing constraints in a learning model may trigger the emergence of grammatical structure as a way to overcome combinatorial problems inherent to language acquisition. It has previously been shown that limited working memory capacity may facilitate grammar induction (Shain et al. 2016). Possibly, models that are able to represent and process very long strings, paired with training on massive data, can achieve what humans judge as grammatically correct output, without establishing a system of grammatical categories and relations.

The training task of LLMs and many other connectionist language learning models is to predict words. In natural language learning, sound or word prediction undoubtedly provides support for efficient language processing, but it is theoretically possible to learn a language without predicting upcoming elements. On the other hand, the identification of units that can be mapped to meanings, is a step that needs to be taken to achieve both understanding and production. While segmentation is not the only or necessarily the most important task in language learning, it is a task that is relatively easy to implement in a model, as cues to rewards are present in text, and it may bring the modeled process closer to a real life learning process than word prediction.

2.3 Parsing and Grammar Induction

Automatic syntactic parsers superficially share our goal of extracting grammatical structure from linguistic input. These models are, however, often not built with the aim of representing cognitively plausible learning processes. Supervised parsing depends on predefined syntactic rules and/or categories, human labeling, and training (Manning and Schutze 1999; Sanford and Sturt 2002; Nivre et al. 2020) and does in that sense not resemble informal language learning.

In contrast, unsupervised grammar induction aims at inferring grammatical structure from linguistic data without any explicit training on how to encode syntactic relationships. Many unsupervised grammar induction models still assume the existence of given formal syntactic relations, providing the model with initial information on categories related to grammatical dependencies or constituents (Muralidaran, Spasić, and Knight 2021). This top–down perspective is not compatible with domain-general learning. Unsupervised data-oriented parsing (DOP) models are more compatible with a usage-based perspective, as they apply a bottom–up methodology with no predefined categories. In this approach, the most probable binary trees are computed from the shortest derivations of sentences (Bod 2006, 2009; Post and Gildea 2013). Another approach that makes relatively few cognitive assumptions is the unsupervised hierarchical hidden Markov models (UHHMM) presented by Shain et al. (2016), which learns hierachical syntax statistically. While DOP and UHHMM demonstrate that grammar can be induced without relying on predefined categories, one drawback of these models lies in their need for simultaneous access to large amounts of data for probability computations. This requirement renders them both cognitively implausible and computationally costly. Another theory-neutral approach is the automatic distillation of structure (ADIOS), a model that extracts statistical patterns incrementally from a corpus of sentences and outputs a directed multigraph, whose patterns can be represented as a context-free grammar (Solan et al. 2003; Berant et al. 2007; Brodsky and Waterfall 2007). Differently from DOP and UHHMM, ADIOS carries out structural inferences locally. However, before extracting statistical patterns, pre-processing of the data is made where the corpus is loaded to a pseudograph with lexical entries as vertices. The local inference is thus not made on a raw stream of linguistic input.

Another paradigm of automatic grammar induction involves assessing hypothetical grammars of formal languages via Bayesian inference (Rule et al. 2018; Rule, Tenenbaum, and Piantadosi 2020; Piantadosi, Tenenbaum, and Goodman 2016; Amalric et al. 2017; Planton et al. 2021; Ullman, Goodman, and Tenenbaum 2012; Ellis et al. 2018). The formal languages are composed of sequences generated by rule systems known as artificial grammars organized within a mathematical hierarchy of escalating generative power, commonly referred to as the Chomsky hierarchy (Chomsky 2002, 1959). The Bayesian approach to formal language learning assumes the existence of a prior that is evaluated against data. This means that the learner needs to formulate and reformulate hypotheses about the entire system that is being learned, overlooking the possibility that grammatical rules may be inferred without a prior framework or without formulating hypotheses about the entire system. Furthermore, many formal languages in such studies do not resemble natural languages. Context-free languages like $a^n b^n$, for example, are even impossible for humans to learn without explicit counting if n is high. Results from Hochmann, Azadpour, and Mehler (2008) suggest that human recognition of such strings relies on distributional regularities rather than on hypotheses about underlying grammar.

We propose a perspective that differs in some crucial aspects from all grammar induction models reviewed here. First, we initially present the learner with a stream of words without cues of sentence borders, whereas other grammar induction studies present sentences one by one. Presenting an unsegmented stream makes the input more similar to that of an early language learner and, at the same time, allows for using sentence borders as cues for reinforcement. Secondly, we apply local decision-making and updating of probabilities of actions, and never let the learner have access to more data than the current state and values of actions associated with it. This is not only computationally cheap, but also an empirically grounded approach to learning. Third and lastly, reinforcement learning allows for structures to be optimized for a task related to language learning instead of being identified purely on the basis of statistical regularities. We are primarily interested in exploring this new approach as a way of understanding how humans induce grammar from data, and not necessarily in optimizing performance in parsing. For this reason we do not compare our model's performance with other grammar induction models at this stage, but rather study its learning process and evaluate its potential for understanding the cognitive prerequisites underlying human language learning.

3. Computational Framework

The method used in this article is that of reinforcement learning applied to a Markov Decision Process (MDP) (Sutton and Barto 2018), that encodes a language learning task. The learner tries to identify sentences by placing boundaries between words and gets positive reinforcement when it succeeds and negative reinforcement when it fails. The model will then be evaluated by looking at learning curves, that is, the fraction of correct responses from a population of agents, and at the grammar induced by the model. This section provides an overview of the learning task, learning algorithms, and evaluation methodology.

3.1 The Learning Task

We start by defining the learning task our cognitive architecture will use. Following Sutton and Barto (2018), we define the task as an MDP. MDPs provide a useful framework for studying learning from interactions to achieve a goal. In this framework, a learner is called an **agent**. In our case, the agent aims to identify sentences in a sequence of words constituting the environment. The sequence of words used here is generated using a probabilistic context free grammar (PCFG), but our model should work on any sequential input with well-defined units. To illustrate the type of input the model receives, it is useful to take a natural language example. Consider, for example, the following three sentences:

The cat chases the dog. The man loves his girlfriend. The sun shines.

Because identifying sentences in this context is too easy due to punctuation and capitalization, we create a raw input by removing any hint as to the beginning of sentences—

for example, capital letters and punctuation are removed from the list of words—yielding the following modified input:

the cat chases the dog the man loves his girlfriend the sun shines

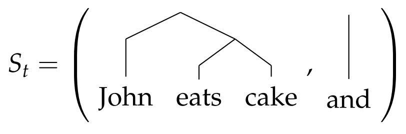
The information about sentence boundaries is stored in a Boolean sequence encoding whether a word in the raw input is preceded by a boundary or not. This Boolean sequence is used to drive the reinforcement procedure. The aim of the learning agent is then to identify sentences, that is, sequences of words separated by boundaries and not containing any other boundaries. This task can be seen as a masking task, where sentence boundaries are masked and the aim of the model is to predict where boundaries occur.

3.1.1 MDP Formalism: States, Actions, and Rewards. In order to specify this task within the MDP framework, we need to specify

1. how agents represent information (i.e., their state space);
2. what they can do in a given state (i.e., their action spaces, which depends on their state), and
3. what consequences their actions have (i.e., the reward structure and transitions between states).

More specifically, the agent interacts with its environment at discrete time steps, $t = 0, 1, 2, 3, \ldots$. At each time step t, the agent is in a given state $S_t \in \mathcal{S}$ and takes an action $A_t \in \mathcal{A}(S_t)$. As a result of that action, the agent receives a reward $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$ and finds itself in a new state $S_{t+1} \in \mathcal{S}$. To encode the behavior of the agent, we use state-action values $Q(s, a)$ encoding the value of taking action $a \in \mathcal{A}(s)$, when the agent finds herself in state s and define the policy of an agent $\pi(a|s)$ as the probability to select action $a \in \mathcal{A}(s)$ when in state s.

3.1.2 Specifying the Language Learning Task as an MDP. In our case, the states are encoded as pairs of structured *chunks* of words. A chunk refers to a sequence of words and an associated binary tree with the words as leaves. For a chunk of length n, there are $C_{n-1} = \frac{1}{n}\binom{2(n-1)}{n-1}$ possible binary tree structures, where C_n is the n-th Catalan number. By default, the second and most recently perceived chunk always consists of a single word. An example state is given by:



(1)

In such a state, the agent has a number of possible actions: She can choose to place a boundary between the two elements of her state, or to integrate the second element into the structure of the first. Figure 1 illustrates the four possible actions associated with the example above. The possible actions in a given state depend on the structure of the binary tree and on how many ways it can be chunked with the second element of the state. We define the right-depth of a state $d(S_t)$ as the number of ways the second

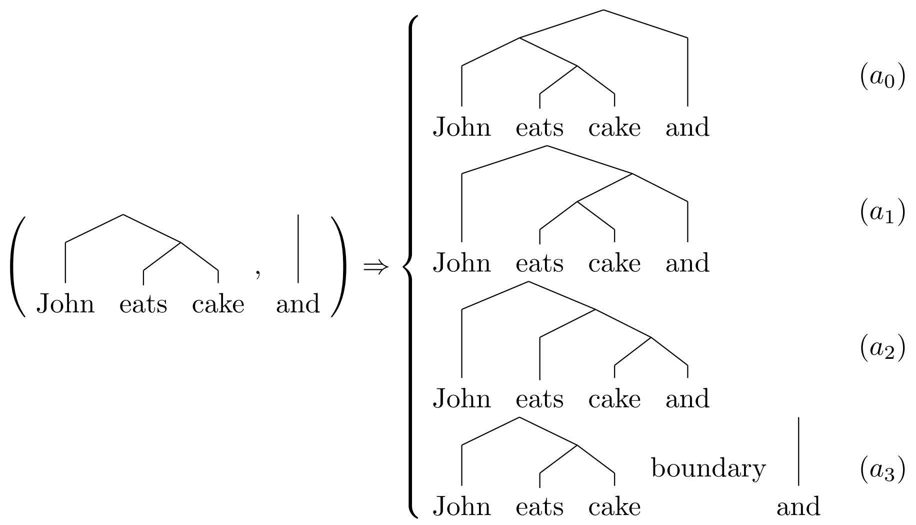


Figure 1
Illustration of the four possible actions. On the right, the results of the four actions are shown. As can be seen, there are three different ways of chunking the second element into the first.

chunk can be integrated into the first. In the example above $d(S_t) = 3$, as is illustrated in Figure 1. There are $d(S_t) + 1$ possible actions $a_0, \dots, a_{d(S_t)}$ in state S_t. We refer to the chunking actions as a_0 when chunking at the root, and we increase the index as we go down the tree. The deepest chunking action we can take is $a_{d(S_t)-1}$. For example, action a_2 in Figure 1 corresponds to grouping the last two stimuli together. The last action $a_{d(S_t)}$ corresponds to placing a boundary between the elements of S_t.

When a chunking action is selected, the resulting chunk becomes the first element of S_{t+1} and the next word is read. In this case, no reward is given.

Upon boundary placement, reinforcement is triggered. Whenever the first element of S_t is a correct sentence, a positive reward of $R_{t+1} = r_+ > 0$ is given. Otherwise, a negative reward of $R_{t+1} = r_- < 0$ is given. The next section describes how the rewards are used to update state-action values and the policy of the agents. After reinforcement, we considered two different ways of reinitializing the task:

Continuous. The second element of S_t becomes the first element of S_{t+1} and the next word is read to instantiate the second element.

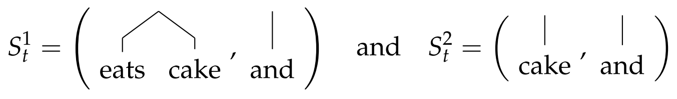
Next sentence. The first element of S_{t+1} is set to the first word of the next sentence in the sequence of words, and the second element is the following word.

These types of reinitialization processes make the MDP episodic in the sense that the process terminates upon boundary placement and restarts in a new state. Thus, each attempt to identify a sentence constitutes a new episode. Note that we are mainly interested in the *Continuous* condition, since this is the only condition where the two borders need to be identified. The *Next sentence* condition corresponds to a simpler task where only the second boundary needs to be identified. We chose to study the *Next sentence* condition as an intermediary condition between the *Continuous* condition and the situation where the two borders are known, common in other grammar induction models.

3.2 Learning Algorithms

During learning, an agent must choose between actions in order to increase its expected reward. In this article, we use a variant of the Q-learning algorithm (Sutton and Barto 2018), in which state-action values are updated according to a temporal difference algorithm. These state-action values can be considered as estimators of the expected reward from that specific state and learning is driven by reducing the error in prediction. In our case, states possess a hierarchic structure whenever $d(S_t) > 1$ and it is possible to modify the standard Q-learning algorithm to take advantage of this structure.

3.2.1 State-action Values, Sub-states, and Sub-actions. In order to decide what to do in a given state, an agent relies on the state-action value $Q(s, a)$ of taking action a in state s. However, the states we consider here have a hierarchic structure. If the right-depth of a state $d(s_t) > 1$, then there are a number of associated sub-states and corresponding actions. Consider again the state in Equation (1). Since this state has right-depth 3, it has two associated sub-states, namely,



where we use an upper index to label these sub-states. The original state is labeled 0 or is left unlabeled. An action $a_i^0 \in \mathcal{A}_t^0(S_t^0)$ associated with the state corresponds to an action $a_{i-k}^k \in \mathcal{A}_t^k(S_t^k)$ associated with sub-state S_t^k whenever $i - k \geq 0$. To specify the learning mechanism, it is convenient to use the Heaviside step function $H(x)$, defined as

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

To take advantage of this structure, we define two composite versions of the state-action values:

1. The **average composite state-action value** $\overline{Q}(s, a_i)$ is defined as:

$$\overline{Q}(s, a_i) := \frac{\sum_{k=0}^{d(s)-1} Q(s^k, a_{i-k}^k) H(i-k)}{\sum_{k=0}^{d(s)-1} H(i-k)}, \quad i = 0, \ldots, d(s)$$

2. The **additive composite state-action value** $\widetilde{Q}(s, a_i)$ is defined as:

$$\widetilde{Q}(s, a_i) := \sum_{k=0}^{d(s)-1} Q(s^k, a_{i-k}^k) H(i-k), \quad i = 0, \ldots, d(s)$$

The average composite state-action value $\overline{Q}(s, a_i)$ averages the state-action values of a state and its sub-states. The Heaviside function selects the actions that are valid for the sub-states. This quantity will be used in decision-making. The additive composite state-action value $\widetilde{Q}(s, a_i)$ computes the sum of state-action values from a state and its associated sub-states. This quantity will be used in the learning mechanism to implement blocking in a similar way as Rescorla and Wagner (1972).

3.2.2 Updating State-Action Pairs. We are now in a position to explain how state-action values are updated. We consider two alternative learning rules:

Q-learning. In this case, state-action values associated with states and sub-states are updated independently. The updating rule is given by

$$Q(S_t^k, a_{i-k}^k) \leftarrow Q(S_t^k, a_{i-k}^k) + \alpha \left[R_T - Q(S_t^k, a_{i-k}^k)\right] H(i-k)$$

Rescorla-Wagner Q-learning. In this case, state-action values associated with states and sub-states are considered elements of a composite stimulus. The updating rule is given by

$$Q(S_t^k, a_{i-k}^k) \leftarrow Q(S_t^k, a_{i-k}^k) + \alpha \left[R_T - \widetilde{Q}(S_t, a_i)\right] H(i-k)$$

In both cases, R_T is the reward obtained upon boundary placement and α is a learning parameter. The update is performed for all time steps $t \in 0, \dots, T$ of the episode and for all sub-states $k \in 0, \dots, d(S_t)$. These temporal difference algorithms are driven by the error in prediction of the reward R_T. The Heaviside function is there to avoid updating an action that does not exist in a given sub-state. The difference between Q-learning and Rescorla-Wagner Q-learning is that in the first case, the state-action values are used as predictors, whereas in the latter case, we use the additive composite state-action values.

State-action values are initialized by setting

$$Q(S_t, a_i) = \begin{cases} q_b & \text{if } i = d(S_t), \\ q_c & \text{otherwise} \end{cases}$$

The parameters q_b and q_c encode the initial values of placing a border or choosing a chunking behavior, respectively.

3.2.3 Using State-Action Pairs to Make Decisions. In order to choose actions in a given state, S_t, we use a softmax policy. The softmax policy computes the probability of choosing a given action using the rule

$$\pi(a_i|S_t) = \frac{e^{\beta \overline{Q}(S_t, a_i)}}{\sum_{k=0}^{d(S_t)+1} e^{\beta \overline{Q}(S_t, a_k)}}$$

where β is a parameter controlling the amount of exploration performed by the model. Note that we use the average composite state-action values $\overline{Q}(S_T, a_i)$ as the support for each action. We want to take advantage of the internal structure of the state and this is the best way to avoid creating biases towards one specific behavior. We use the same policy for the two possible learning rules mentioned above. Overall, the model has 6 parameters, summarized in Table 1. The values of these parameters have been chosen to initially favor boundary placement ($q_b > q_c$) in order to avoid the model chunking

Table 1
Parameters of the model.

Parameter	Interpretation	Value
α	Learning rate	0.1
β	Exploration parameter	1.0
r_+	Positive reward	25
r_-	Negative reward	-10
q_b	Initial value for border	1
q_c	Initial value for chunking	-1

everything, which would inhibit learning. The values of the other parameters have been chosen to obtain reasonable learning times. However, there is room for optimization. Our aim is not to get the best possible performances, but rather to obtain a proof of principle that our architecture works as intended.

3.3 Evaluation Methods

To evaluate the performance of the model, we need to assess whether it successfully learns the input languages and, if so, how quickly. In addition, we are interested in what grammatical information it extracts and to study the process of grammar induction. To achieve these goals, we use learning curves to test the ability of the model to learn and quantify its speed. Grammatical information is extracted directly from the state-action values at different stages of the learning process and tested at the same time points.

3.3.1 Extracting Learning Curves. In order to get a learning curve, we simulate N agents and record for each episode of the MDP whether the decision was correct or incorrect. We then compute the fraction of agents being successful at each episode and plot the fraction of correct responses as a function of the number of episodes of the MDP that we refer to as **trials**. If all agents eventually learn the language perfectly, then the fraction of correct responses should go to one.

In order to estimate the learning speed, we fit a logistic curve of the form

$$f(x) = \frac{1}{1 + e^{-k(x-x_0)}} \tag{2}$$

which has the two parameters k and x_0. k is a measure of the stiffness of the logistic curve and x_0 corresponds to the parameter where the function value is $\frac{1}{2}$. We estimate the learning time as $2x_0$, since at the beginning of the simulation the fraction of correct responses is 0.

3.3.2 Testing the Performance of the Model. Since we are interested in how learning unfolds, we need to test the performance of the model at various stages of the learning process. To this end, we use a test sequence consisting of n_{test} sentences produced with the same PCFG as the word sequences used for learning. Due to the small size of the languages studied, and the lack of strong generalization, sentences in the test sequence overlap with the sentences used in the training phase. We do not consider this to be problematic, since the beginning and the end of these sentences must be correctly identified for the system to perform well on the test sequence.

To evaluate how learning unfolds, we pause learning at different times of the learning process and record the performance of the model on the test sequence without learning. The learning phase then continues until the next snapshot time is reached. We measure the fraction of correct sentence identification and qualitatively discuss the tree structures used by the model for these sentences. In particular, we discuss the distribution of tree structures associated with a given sentence structure and the changes of that distribution over time.

3.3.3 Extracting Grammatical Information. In our model, all grammatical information is encoded in the state-action values of states and their associated sub-states, namely, in the Q values and the $\overline{Q}$ values. In order to extract grammatical information, we need to extract the most likely action to be taken in each state. This information is encoded in the maximum value of $\overline{Q}$. Furthermore, the states used by the model are lexicalized, that is, individual words are not grouped into grammatical categories. However, we can group states sharing the same structure according to the underlying PCFG to display the results. As a proxy for the learned information, we choose a threshold $0 < t_G < r_+$ and extract all state-action values greater than t_G and report averages of Q and $\overline{Q}$ over lexicalized states sharing the same structure. Displaying both Q and $\overline{Q}$ is informative because if $Q > \overline{Q}$, then the larger structure controls the decision, whereas when $Q < \overline{Q}$ the decision is controlled by sub-states. Of course, state-action with lower values also constitute grammatical information, but a threshold is necessary to select data to display. For each state (i.e., for each pair of chunks) we then have the most likely action to be taken and its associated values. This representation of grammatical information does not constitute a standard grammar, but is instructive for understanding how an agent learns. The extracted grammatical information is then saved in an Microsoft Excel spreadsheet. The code of this project as well as all Excel files used in this project are available online in the GitHub repository: `https://github.com/michaudj/LanguageLearner/`.

Because one of the aims of this article is to provide a transparent model of language acquisition, we use this technique of extracting grammatical information at various stages of the learning process and save them as different sheets in an Excel file. This allows us to look at how the grammatical information of an agent changes over time and whether it converges to a systematic representation.

For more complex languages in which the number of relevant state-action values is too high to be displayed, we will provide relevant examples and a more qualitative discussion of the learned grammatical information.

4. Results

In this section, we present the results of our experiments on various artificial languages. We start by considering a simple language consisting of nouns (N) and verbs (V) organized in sentences with the structure noun-verb-noun (NVN), and then move on to consider more complex grammars. The NVN language is agnostic to the subject and object status of the nouns and may thus represent sentences occurring in English as well as many other languages. In the more complex grammars we use English as a model language but focus on phenomena that are typical of many languages with diverse typological profiles, like recursion and variable multi-word constituents.

4.1 A Baseline Case: NVN Languages

Our baseline case has been chosen to illustrate how the agent learns the language and will be used to motivate the choice of learning algorithm used for more complex languages. We start by considering a simple NVN language. In this language, word transitions are enough to identify sentences, which, among other things, allows for testing whether the model is able to identify the most economic strategy of learning.

Our language is generated by PCFG $G = (\mathcal{M}, \mathcal{T}, \mathcal{R}, \mathcal{S}, \mathcal{P})$, where $\mathcal{M}$ is the set of non-terminal symbols, $\mathcal{T}$ is the set of terminal symbols, $\mathcal{R}$ is the set of production rules, $\mathcal{S}$ is the start symbol, and $\mathcal{P}$ is the set of probabilities on production rules.

4.1.1 Defining the Grammar. For our simple NVN grammar, we use

$$\begin{aligned}\mathcal{M} &= \{S, N, V\}\\ \mathcal{T} &= \{n_1, n_2, \ldots, n_{K_n}\} \cup \{v_1, v_2, \ldots, v_{K_v}\}\\ \mathcal{S} &= S\end{aligned}$$

where K_n is the total number of nouns and K_v is the total number of verbs. The production rules are:

$$R = \left\{\begin{array}{c} S \rightarrow N, V, N \\ N \rightarrow n_1|\; n_2|\; n_3|\ldots \\ V \rightarrow v_1|\; v_2|\; v_3|\ldots \end{array}\right\} \tag{3}$$

and are equiprobable. This means that S always rewrites as N, V, N, and the nouns N rewrite as n_i, $i \in [1, 2, \ldots, K_n]$ uniformly at random, and similarly for the verbs.

4.1.2 Learning Curves. With this NVN language defined, we test the ability of our models to learn this language using the proposed task. We consider the four cases combining the definition of Sections 3.1.2 and 3.2.2:

QC. Q-learning with continuous border condition;

QN. Q-learning with next sentence border condition;

RWQC. Rescorla-Wagner Q-learning with continuous border condition;

RWQN. Rescorla-Wagner Q-learning with next sentence condition.

The obtained learning curves are displayed in Figure 2. We see that all variants of the model successfully learn the language reaching a fraction of correct responses very close to 1. Using the fit to the logistic curve in Equation (2), we estimated the learning times of the four models in Table 2.

We observe that when the next sentence condition is used, learning is faster. This is not surprising, since the task is easier because only one of the two boundaries of a sentence has to be identified. With respect to learning times, there is very little difference between Q-learning and Rescorla-Wagner Q-learning, which is expected since the two learning algorithms are very similar. The main difference between these two algorithms is the information used to drive error-correction, namely, whether blocking is used or not. This will be examined below when we look at what is learned at different stages of the learning process.

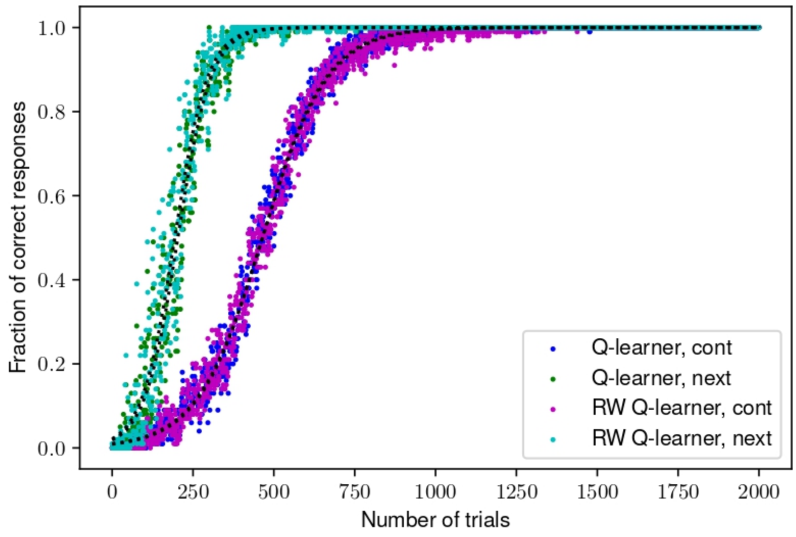


Figure 2
Learning curves for the four combinations of sentence conditions QC, QN, RWQC, and RWQN. The NVN language has $K_n = K_v = 5$. Fractions are obtained over 100 agents.

Table 2
Learning times of the four learning algorithms for a NVN language with $K_n = K_v = 5$.

Algorithm	**Learning time**
QC	927
QN	402
RWQC	925
RWQN	388

4.1.3 Grammar Extraction and Acquisition Process. As we have seen above, the performance of Q-learning and Rescorla-Wagner Q-learning are very similar in terms of learning curve. The question addressed in this section is what type of grammatical information is extracted and used by these two learning algorithms and whether the sentence condition affects what is learned. To achieve this, we report the grammatical information extracted in terms of the two elements of a state/sub-state, the action with the highest state-action value and the corresponding average composite state-action value. The results are shown in Tables 4 and 5 for the continuous border condition and are extracted from the two Excel spreadsheets QLearnerC2_revisedFinal.xlsx and RWQLearnerC2_revisedFinal.xlsx available in the GitHub repository: `https://github.com/michaudj/LanguageLearner/`. Snapshots are taken at the beginning of the S-shaped learning curve, when learning fraction is close to 0.5, when learning is just completed, and at two later points, when performances are very high. The columns labeled # refer to the number of lexicalized states (s_1, s_2) matching the pattern. For example, two states $([n_2, v_3], n_4)$ and $([n_1, v_5], n_2)$ have the same structure and constitute two instances of the $([N, V], N)$ state. The results for the next sentence condition are similar and can be found in Excel spreadsheets QLearnerN2_revisedFinal.xlsx and RWQLearnerN2_revisedFinal.xlsx in the GitHub repository.

Our results show that the Rescorla-Wagner algorithm is much more parsimonious in the number of state-action values it relies upon. This is best seen by looking at the number of instances of a given state-action pair reported in column "#" of Tables 4 and 5. While these numbers steadily increase in Q-learning, only some of them increase in Rescorla-Wagner Q-learning and some of them even decrease after some time. For example, in the last three rows of Table 5. This shows that the model learns more specific information than the Q-learning algorithm. The same can be observed in the averaged state-action values, where all actions reported an increase in value for the Q-learning algorithm, while only some of them increase for Rescorla-Wagner Q-learning and some even decrease after some point. In addition, the effect of blocking is clearly visible when looking at the last three rows, since the value of the average composite state-action $\overline{Q}$ does not decrease, showing that the sub-state are contributing more to the decision than the state itself.

The results for Q-learning suggest that all state-action pairs will eventually reach the maximum reinforcement value of 25 and the rule will not specify a preferable tree structure over the sentence. For instance, both $[N, [V, N]]$ and $[[N, V], N]$ tree structure will occur with significant probability. This grammatical output is not very informative. In order to test this hypothesis, we conduct a performance test at the same snapshot times as for the Q values reported in Tables 4 and 5 using a test sequence of length $n_{\text{test}} = 100$. The results are reported in Table 3. We see that in the Rescorla-Wagner Q-learning, the number of sentences with the tree structure $[N, [V, N]]$ is larger than for Q-learning. In addition, the next sentence condition produces a large bias towards that same structure. This is likely due to the fact that this specific structure has a higher right-depth and recruits information from more sub-states in the decision-making process. As argued above, we are providing results for the next sentence condition mainly for comparison, since we are interested in the full segmentation task. Together, these results suggest that the Rescorla-Wagner version of the algorithm is a better choice.

Table 3
Test results at different stages of learning for the four models QC, QN, RWQC, and RWQN. The test consists of $n_{\text{test}} = 100$ sentences. The number reported are the number of correctly identified sentences in the test set associated with the corresponding tree structure. When the total number does not add up to 100, it means that the other sentences were not correctly identified.

		Snapshot times				
Model	Tree	300	600	900	2,000	4,000
QC	$[[N, V], N]$	10	15	44	45	45
	$[N, [V, N]]$	10	25	51	55	55
	Total	20	40	95	100	100
QN	$[[N, V], N]$	30	21	18	18	18
	$[N, [V, N]]$	55	78	82	82	82
	Total	85	99	100	100	100
RWQC	$[[N, V], N]$	4	34	31	36	36
	$[N, [V, N]]$	5	20	50	64	64
	Total	9	54	81	100	100
RWQN	$[[N, V], N]$	26	15	15	15	15
	$[N, [V, N]]$	66	85	85	85	85
	Total	92	100	100	100	100

Table 4
Grammar extracted for a NVN language with $R_n = R_v = 5$ by the Q-learning algorithm with continuous border. Snapshots are taken after 300, 600, 900, 2,000, and 4,000 trials and a threshold of $t_G = 5$. Columns labeled "$Q/\overline{Q}_i$" report averaged values of $Q((s_1, s_2), a_j)$ and $\overline{Q}((s_1, s_2), a_j)$ over the number of relevant instances (reported in the # columns) after i trials. There are two values whenever $Q \neq \overline{Q}$. The total number column reports the maximum number of lexicalized states matching the pattern.

s_1	s_2	action	$Q/\overline{Q}_{300}$	#	$Q/\overline{Q}_{600}$	#	$Q/\overline{Q}_{900}$	#	$Q/\overline{Q}_{2000}$	#	$Q/\overline{Q}_{4000}$	#	Total
N	N	a_1 (border)	6.76	1	6.18	4	10.90	21	24.78	25	25.00	25	25
N	V	a_0 (chunk)	7.94	1	6.59	5	10.13	22	24.77	25	25.00	25	25
V	N	a_0 (chunk)	6.81	1	7.05	2	11.22	7	22.89	17	24.58	18	25
$[N, V]$	N	a_0 (chunk at root)	—	—	8.03	1	6.99	17	16.36	55	23.26	53	125
$[N, V]$	N	a_1 (chunk deep)	7.94/7.38	1	7.69/6.32	3	6.99/8.62	17	14.80/19.27	69	22.90/23.89	72	125
$[V, N]$	N	a_2 (border)	—	—	6.40/5.04	6	7.08/9.71	20	13.87/19.32	81	20.90/22.95	88	125
$[[N, V], N]$	N	a_2 (border)	—	—	5.56/1.81	1	6.01/9.38	13	7.44/16.12	186	11.08/18.04	267	625
$[N, [V, N]]$	N	a_3 (border)	—	—	5.56/5.45	6	5.77/8.61	18	7.10/15.83	197	10.96/19.41	344	625

Table 5
Grammar extracted for an NVN language with $R_n = R_v = 5$ by the RW Q-learning algorithm with continuous border. Snapshots are taken after 300, 600, 900, 2,000, and 4,000 trials and a threshold of $t_G = 5$. Columns labeled "$Q/\overline{Q}_i$" report averaged values of $Q((s_1, s_2), a_j)$ and $\overline{Q}((s_1, s_2), a_j)$ over the number of relevant instances (reported in the # columns) after i trials. There are two values whenever $Q \neq \overline{Q}$. The total number column reports the maximum number of lexicalized states matching the pattern.

s_1	s_2	action	$Q/\overline{Q}_{300}$	#	$Q/\overline{Q}_{600}$	#	$Q/\overline{Q}_{900}$	#	$Q/\overline{Q}_{2000}$	#	$Q/\overline{Q}_{4000}$	#	Total
N	N	a_1 (border)	—	—	7.16	3	12.66	23	19.72	25	20.47	25	25
N	V	a_0 (chunk)	—	—	7.11	4	13.97	24	24.81	25	25.00	25	25
V	N	a_0 (chunk)	—	—	6.98	3	12.13	16	19.98	19	20.23	19	25
$[N, V]$	N	a_0 (chunk at root)	—	—	6.05	3	6.80	18	14.84	43	21.97	43	125
$[N, V]$	N	a_1 (chunk deep)	5.59/3.57	1	7.36/6.32	2	6.28/9.74	19	6.18/12.94	31	6.54/12.59	14	125
$[V, N]$	N	a_2 (border)	—	—	5.96/3.96	5	6.36/9.45	23	5.51/12.16	28	5.52/12.16	4	125
$[[N, V], N]$	N	a_2 (border)	—	—	5.63/3.51	3	5.73/9.36	6	5.52/12.95	11	5.31/12.91	9	625
$[N, [V, N]]$	N	a_3 (border)	—	—	5.73/4.02	5	5.81/8.25	8	5.70/9.84	5	5.14/10.02	2	625

The results for Rescorla-Wagner Q-learning suggest that if learning continues only 4 state-action pairs are relevant for the decision-making process:

1. Place border between N and N;
2. Chunk together N and V;
3. Chunk together V and N;
4. When $[N, V]$ is followed by N, chunk at the root in 43/125 cases, yielding $[[N, V], N]$ and chunking deep otherwise, yielding $[N, [V, N]]$, in good agreement with the test performances provided in Table 3.

This means that all information about sentence boundary placement is encoded in the $N - N$ transitions and rules 2, 3, and 4 control the emergent tree structures. This model seems much more informative about the learning process and is, in some sense, minimalist with regard to the information it extracts. Therefore, we choose Rescorla-Wagner Q-learning for the analysis of more complex sentence structures.

4.2 Towards Natural Language Input

In this section, we explore how the Rescorla-Wagner Q-learning algorithm with continuous border condition performs on more complex grammars. We consider a number of different grammars. First, we consider a grammar that has both mono- and ditransitive verbs to explore the model's behavior and performance when introducing a minimal change that makes $N - N$ transitions unreliable predictors of sentence transitions. We then increase the complexity of the grammars by either introducing relative clauses or a number of other adnominal elements. Relative clauses allow for including recursive repetition of structures within a sentence, a very common feature of the world's languages. Complexifying the noun phrase allows for studying how the models behave when exposed to variable structures within a recurring constituent.

For the two latter more complex languages, the grammatical information extracted will often be too big to analyze in a table as we have done for the NVN language. Instead, we study the learning curve and break it down by sentence length to get some insight into which sentences are learned first. This analysis is complemented by performance tests. We then examine some examples of tree structures to show some systematic reuse of identified structures.

4.2.1 Mono- and Ditransitive Verb. We first consider a grammar with both monotransitive verbs (MV), that are always followed by one noun, and ditransitive verbs (DV), that are always followed by two nouns. The grammar is defined as follows:

$$\begin{aligned}
\mathcal{M} &= \{S, N, MV, DV\},\\
\mathcal{T} &= \{n_1, n_2, \ldots, n_{K_n}\} \cup \{mv_1, mv_2, \ldots, mv_{K_m}\} \cup \{dv_1, dv_2, \ldots, dv_{K_d}\},\\
\mathcal{S} &= S,
\end{aligned}$$

where, K_n is the total number of nouns, K_m is the total number of monotransitive verbs, and K_d is the total number of ditransitive verbs. The production rules are

$$\mathcal{R} = \left\{ \begin{array}{r} S \rightarrow N, MV, N|\ N, DV, N, N \\ N \rightarrow n_1|\ n_2|\ n_3|\dots \\ MV \rightarrow mv_1|\ mv_2|\ mv_3|\dots \\ DV \rightarrow dv_1|\ dv_2|\ dv_3|\dots \end{array} \right\} \tag{4}$$

and are equiprobable. This means that S always rewrites as N, MV, N in 50% of the cases and as N, DV, N, N otherwise. The different word classes rewrites uniformly as in the NVN language above. We refer to this language as the MD language.

Here, we report results for a MD language with $K_n = 5$, and $K_m = K_d = 1$. We simulate the learning using the Rescorla-Wagner Q-learning with continuous border condition.

Figure 3 reports the learning curve for this language. The first panel displays the standard learning curve and the second panel shows the learning curve for different sentence lengths. Sentences of length 3 are monotransitive sentences and sentences of length 4 are ditransitive sentences. As can be seen, even when $N - N$ transitions are not a reliable predictor of sentence boundaries, the learning algorithm still learns to identify sentences. The second panel shows that monotransitive sentences are learned first and that there is some interference between the mono- and ditransitive sentences. At the beginning, monotransitive sentences are quickly learned, but the learning slows down as the learner figures out how to deal with ditransitive sentences. After a transition period, both types of sentences are learned. In order to better understand how the processing is done, we look at snapshots taken during the learning process. All data is available online in the Excel spreadsheet RWQLearnerC_MD_revisedFinal.xlsx stored in the GitHub repository: `https://github.com/michaudj/LanguageLearner/`. The results are displayed in Table 6.

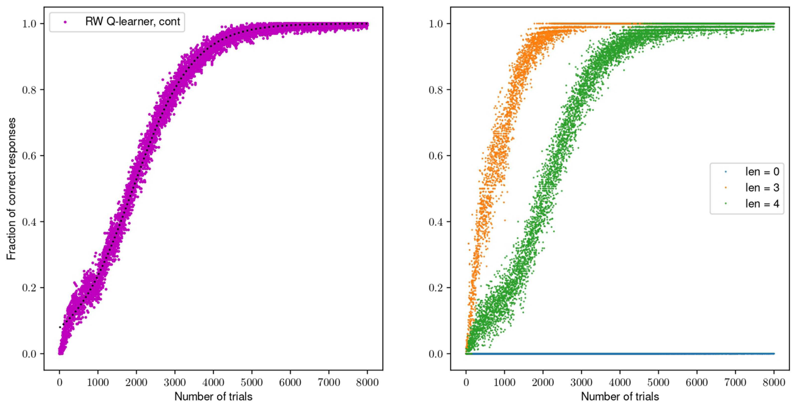


Figure 3
Left: Learning curve for the Rescorla-Wagner Q-learning with continuous border condition for the MD language with $K_n = 5$, $K_m = K_d = 1$. Fractions are obtained over 200 agents. Right: breakdown of the learning curve by sentence length. Note that if at a given trial no learners encounter a sentence of a given length, then it contributes to sentences of length 0, which are, therefore, meaningless.

Table 6
Grammar extracted for an MD language with $R_n = 5$, and $R_m = R_d = 1$ by the RW Q-learning algorithm with continuous border. Snapshots are taken after 1,000, 1,500, 3,000, 6,000, and 25,000 trials and a threshold of $t_G = 10$. Columns labeled "$Q/\overline{Q}_i$" report averaged values of $Q((s_1, s_2), a_j)$ and $\overline{Q}((s_1, s_2), a_j)$ over the number of relevant instances (reported in the # columns) after i trials. There are two values whenever $Q \neq \overline{Q}$. The total number column reports the maximum number of lexicalized states matching the pattern.

s_1	s_2	action	$Q/\overline{Q}_{1000}$	#	$Q/\overline{Q}_{1500}$	#	$Q/\overline{Q}_{3000}$	#	$Q/\overline{Q}_{6000}$	#	$Q/\overline{Q}_{25000}$	#	Total
N	N	a_1 (border)	—	—	—	—	13.02	17	16.50	24	18.08	25	25
N	MV	a_0 (chunk)	—	—	14.05	5	24.92	5	24.42	5	25.00	5	5
N	DV	a_0 (chunk)	—	—	—	—	20.68	5	24.30	5	25.00	5	5
MV	N	a_0 (chunk)	—	—	11.98	2	15.97	5	15.35	5	15.75	5	5
DV	N	a_0 (chunk)	—	—	—	—	18.14	4	19.78	4	19.78	4	5
$[N, MV]$	N	a_0 (chunk at root)	—	—	11.18	4	22.93	8	24.94	8	24.96	8	25
$[N, MV]$	N	a_1 (chunk deep)	—	—	12.85/12.96	1	10.65/12.74	5	10.28/12.10	4	10.51/12.50	8	25
$[N, DV]$	N	a_0 (chunk at root)	—	—	—	—	13.38	4	24.75	4	25.00	4	25
$[N, DV]$	N	a_1 (chunk deep)	—	—	—	—	—	—	11.77/8.82	1	15.44/12.48	1	25
$[MV, N]$	N	a_2 (border)	12.50/4.68	2	13.19/6.15	8	12.46/10.58	15	—	—	—	—	25
$[DV, N]$	N	a_0 (chunk at root)	—	—	—	—	17.69	11	20.37	15	19.38	17	25
$[DV, N]$	N	a_1 (chunk deep)	—	—	—	—	13.05/9.60	4	13.65/10.59	6	14.02/10.41	6	25
$[[N, MV], N]$	N	a_2 (border)	—	—	11.41/5.87	4	12.35/10.81	23	11.55/13.40	20	—	—	125
$[[N, DV], N]$	N	a_0 (chunk at root)	—	—	—	—	12.05	7	19.44	15	24.37	15	125
$[[N, DV], N]$	N	a_1 (chunk deep)	—	—	—	—	—	—	12.83/10.17	3	15.01/12.48	3	125
$[N, [DV, N]]$	N	a_0 (chunk at root)	—	—	—	—	12.56	2	19.69	11	24.44	13	125
$[N, [DV, N]]$	N	a_1 (chunk deep)	—	—	—	—	—	—	10.32/11.67	1	12.33/12.50	3	125
$[N, [DV, N]]$	N	a_2 (chunk deepest)	—	—	—	—	—	—	10.22/8.30	1	10.26/8.33	1	125

The first two snapshots show that at this stage of learning, only information about monotransitive verbs has been learned and decision making relies a lot on full sentences ($Q \geq \overline{Q}$). However, from trial 3,000, the learner starts to rely more on shorter chunks in the decision making process. At this point, the learner starts to have knowledge about ditransitive verbs. At the next snapshot, learning is complete and we see that now boundary placement essentially relies on $N - N$ transitions as in the NVN case. The ditransitive case is solved by using longer chunks containing ditransitive verbs to inhibit border placement and favor a chunking action instead. But in both sentence structures, border placement relies on $N - N$ transitions. This is best seen in the last snapshot where only the $N - N$ transition supports border placements.

In order to see how the performance of the model evolves, we perform a performance test at each of the snapshot times. The test set contains 100 sentences divided into 52 monotransitive sentences and 48 ditransitive sentences. Results of these tests are reported in Table 7. We observe that the tree structures of monotransitive sentences are similar to those of the NVN language, with a bias towards the $[N, [V, N]]$ structure. In contrast, the ditransitive sentences do not have a uniform distribution of tree structures; instead the structure $[N, [[DV, N], N]]$ is overly represented and accounts for 48% of the cases, while the structure $[[N, DV], [N, N]]$ only occurs in 2% of the cases. This suggests that the chunk $[DV, N]$ is used to inhibit border placement and promotes chunking instead, while allowing the final N to correctly predict border placement.

4.2.2 Adding Relative Clauses. The next sentence structure we analyze are relative clauses. We add possible relative clauses to the MD grammar above, by letting some nouns be followed by a relative pronoun (Rel), a monotransitive verb, and a noun. We choose to only include monotransitive verbs in relative clauses to avoid a combinatorial explosion in the output. The grammar is defined as follows:

$$\mathcal{M} = \{S, VP, Rel, N, MV, DV, R\}$$
$$\mathcal{T} = \{n_1, n_2, \ldots, n_{K_n}\} \cup \{mv_1, mv_2, \ldots, mv_{K_m}\} \cup \{dv_1, dv_2, \ldots, dv_{K_d}\} \cup \{r_1, r_2, \ldots, r_{K_r}\}$$
$$\mathcal{S} = S$$

Table 7
Test results for the MD language. The test set contains 100 sentences divided into 52 monotransitive and 48 ditransitive sentences. The table reports the sentences that were correctly identified at each snapshot and the tree structure associated with it. When the total number does not add up to 100, it means that the other sentences were not correctly identified.

		Snapshot times				
Sentence	Tree	1,000	1,500	3,000	6,000	25,000
N, MV, N	$[[N, MV], N]$	12	17	22	23	23
	$[N, [MV, N]]$	12	26	29	29	29
	Total	24	43	51	52	52
N, DV, N, N	$[[[N, DV], N], N]$	4	6	8	9	9
	$[[N, [DV, N]], N]$	1	2	8	7	7
	$[[N, DV]], [N, N]]$	0	2	0	1	1
	$[N, [[DV, N], N]]$	0	4	21	23	23
	$[N, [DV, [N, N]]]$	0	2	9	8	8
	Total	1	10	38	39	39

where, K_n is the total number of nouns, K_m is the total number of monotransitive verbs, K_d is the total number of ditransitive verbs, and K_r is the number of relative pronouns. The production rules are:

$$\mathcal{R} = \left\{ \begin{aligned} S &\rightarrow N, VP \\ VP &\rightarrow MV, N|\ DV, N, N|\ MV, N, Rel|\ DV, N, N, Rel \\ Rel &\rightarrow R, MV, N \\ N &\rightarrow n_1|\ n_2|\ n_3| \dots \\ MV &\rightarrow mv_1|\ mv_2|\ mv_3| \dots \\ DV &\rightarrow dv_1|\ dv_2|\ dv_3| \dots \\ R &\rightarrow r_1|\ r_2|\ r_3| \dots \end{aligned} \right\} \tag{5}$$

and the probabilities of the different production rules for the verb phrase VP are set to 0.3 for production rules without relative clause and to 0.2 for production rules with relative clause; that is, N, MV, N occur 30% of the time, N, DV, N, N occur 30% of the time, N, MV, N, R, MV, N occur 20% of the time, and N, DV, N, N, R, MV, N occur 20% of the time. Other production rules are equiprobable. We refer to this language as the RelClause language.

Here, we report results for a RelClause language with $R_n = 5$, $R_m = R_d = R_r = 1$. Once again, learning is done using the Rescorla-Wagner Q-learning algorithm with continuous border condition. The learning curves are displayed in Figure 4. Once again,

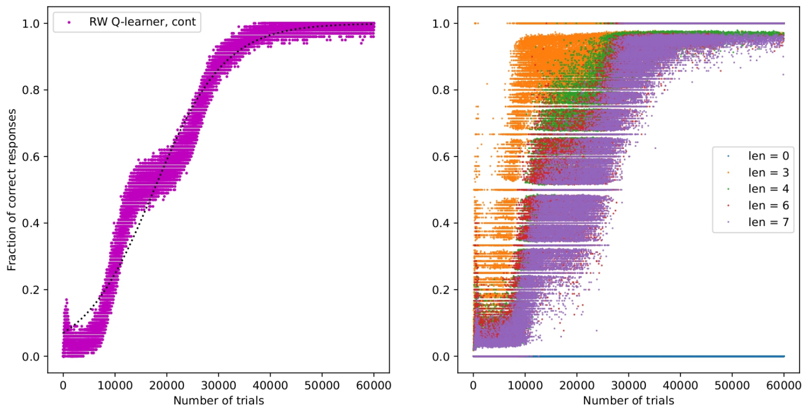


Figure 4
Left: Learning curve for the Rescorla-Wagner Q-learning with continuous border condition for the relative clause language with $K_n = 5$, $K_m = K_d = K_r = 1$. Fractions are obtained over 100 agents. Right: breakdown of the learning curve by sentence length. Note that if at a given trial no learners encounter a sentence of a given length, then it contributes to sentences of length 0, which are, therefore, meaningless. Since only a fraction of learners encounter a sentence of the same length, explaining the layering structure of the breakdown plot.

we see that the model learns this language without problems. Interestingly, we observe a U-shaped curve at the beginning of the learning process and we can clearly see that the learning process is irregular and shows a staircase pattern with a plateau around trial 15,000. The breakdown into sentence lengths is showing that at first short sentences are learned well, but then performances decrease while the system learns to deal with ditransitive structures and relative clauses. After some delay all longer sentences get learned without significant delays, pointing out the generalization process over the end of sentences, for instance, 75% of the sentences end with MV, N, making $[MV, N] - N$ transitions a good predictor of sentence boundaries.

We now take a closer look at the development of the grammatical information of a single learner by extracting snapshots of its state-action values. Selected results are displayed in Table 8. Not all constructions are displayed due to their high number. The display is limited to s_1 constructions of a maximum length of 3 to avoid a combinatorial explosion in the table. The display is also limited to constructions that are relatively frequent or that reach relatively high Q-values, but not necessarily high $\overline{Q}$ values. The threshold of the Q-values of the displayed structure is set to $t_G = 4$, lower than previous thresholds, in order to capture what is going on in the initial U-curve. Full data on all the state-action values above for this learner are available in the Excel file RWQLearnerC_rel_revisedFinal.xlsx available online in the GitHub repository `https://github.com/michaudj/LanguageLearner/`.

In the snapshots, we first explore the initial U-shaped curve using both the Q and $\overline{Q}$ values and performance tests. From the performance test, we see that monotransitive sentences have the structure $[[N, MV], N]$ and Table 8 suggests that this identification relies on the full structure, since only the border placement with that structure has a weight about the threshold. Note that $Q > \overline{Q}$, which suggests that the decision is not based on sub-states. The performance test shows that at this stage some ditransitive sentences are recognized, but without a preferred tree structure. Soon, at 1,000 trials, the model performances deteriorate and the number of sentences correctly identified drops to nearly 0. In Table 8, we observe that the model still relies on full structures for decision-making, but the weights are lower than in the first snapshot. Such a decrease in performances can be explained by the presence of ditransitive sentences in which the $N - N$ transition is not a good predictor of sentence boundaries. After 7,000 trials the U-curve has turned and the model learns that monotransitive sentences should be analyzed $[N, [MV, N]]$ as shown in Table 9 and the $[MV, N] - N$ transition becomes a better boundary predictor. Table 8 also contains more information about ditransitives, but the learner still doesn't know how to process relative clauses. The performances then steadily increase until the learner is proficient. In the specific learner analyzed in detail here, a high proficiency level is only reached at the last snapshot. This is illustrated by the much higher performance on the test set and the much more systematic structures used by the model.

Between trial 15,000 and trial 50,000, the model learns to identify the relative constructions and starts relying on the shorter transitions to a higher degree. This is shown by the better nearly perfect performances of the model and by the Q-values displayed in Table 8. Note that now the $\overline{Q}$ values tend to be higher than the Q values, showing that the model now relies on shorter structures to make decisions. Note that the border is now mainly predicted by three sub-states: $N - N$ transitions, $[MV, N] - N$ transitions, and $[N, N] - N$ transitions. The last two are being used to differentiate between ditransitive sentences and other sentences.

Table 8
Grammar extracted for a RelClause language with $R_n = 5$, and $R_m = R_d = R_r = 1$ by the RW Q-learning algorithm with continuous border. Snapshots are taken after 400, 1,000, 7,000, 15,000, and 50,000 trials, and a threshold of $t_G = 4$. Columns labeled "$Q/\overline{Q}_i$" report averaged values of $Q((s_1, s_2), a_j)$ and $\overline{Q}((s_1, s_2), a_j)$ over the number of relevant instances (reported in the # columns) after i trials. The total number column reports the maximum number
of lexicalized states matching the pattern.

s_1	s_2	action	$Q/\overline{Q}_{400}$	#	$Q/\overline{Q}_{1000}$	#	$Q/\overline{Q}_{7000}$	#	$Q/\overline{Q}_{15000}$	#	$Q/\overline{Q}_{50000}$	#	Total
N	N	a_1 (border)	—	—	—	—	—	—	—	—	13.44	25	25
N	MV	a_0 (chunk)	—	—	—	—	—	—	—	—	21.96	5	5
N	DV	a_0 (chunk)	—	—	—	—	—	—	—	—	24.97	5	5
N	R	a_0 (chunk)	—	—	—	—	—	—	—	—	14.95	4	5
MV	N	a_0 (chunk)	—	—	—	—	—	—	—	—	21.56	5	5
DV	N	a_0 (chunk)	—	—	—	—	—	—	—	—	15.51	5	5
R	MV	a_0 (chunk)	—	—	—	—	—	—	—	—	23.25	1	1
$[N, MV]$	N	a_0 (chunk at root)	—	—	—	—	—	—	4.68	1	25.00	1	25
$[N, MV]$	N	a_1 (chunk deep)	—	—	—	—	7.28/−0.27	5	10.07/0.00	24	4.36/12.35	7	25
$[N, DV]$	N	a_1 (chunk deep)	—	—	—	—	—	—	5.12/−2.11	4	9.47/12.49	25	25
$[N, N]$	N	a_2 (border)	—	—	—	—	5.37/−1.05	2	5.65/−2.68	8	5.16/9.05	69	125
$[R, MV]$	N	a_1 (chunk deep)	—	—	—	—	—	—	—	—	4.56/13.00	3	5
$[MV, N]$	N	a_2 (border)	—	—	—	—	7.70/−0.79	5	8.28/−1.20	24	7.32/10.38	25	25
$[MV, N]$	R	a_0 (chunk at root)	—	—	—	—	—	—	—	—	19.72	2	5
$[MV, N]$	R	a_1 (chunk deep)	—	—	—	—	—	—	—	—	7.76/11.33	3	5
$[DV, N]$	N	a_0 (chunk at root)	—	—	—	—	—	—	—	—	19.77	7	25
$[DV, N]$	N	a_0 (chunk deep)	—	—	—	—	—	—	5.21/−1.13	3	18.87/10.43	18	25
$[[N, MV], N]$	N	a_2 (border)	5.34/0.90	1	4.30/−0.76	5	5.71/0.41	24	9.12/0.22	16	7.86/10.82	17	125
$[N, [MV, N]]$	N	a_3 (border)	—	—	4.22/0.63	1	5.88/−1.16	11	9.11/2.14	98	4.38/8.42	99	125
$[N, [MV, N]]$	R	a_1 (chunk deep)	—	—	—	—	—	—	—	—	5.24/12.48	10	25
$[N, [MV, N]]$	R	a_2 (chunk deepest)	—	—	—	—	—	—	5.32/−1.45	1	—	—	125
$[N, [DV, N]]$	N	a_1 (chunk deep)	—	—	—	—	—	—	4.49/1.12	1	5.22/12.51	29	125
$[N, [DV, N]]$	N	a_2 (chunk deepest)	—	—	—	—	—	—	5.36/0.40	5	4.80/8.53	53	125
$[MV, [N, R]]$	MV	a_1 (chunk deep)	—	—	—	—	—	—	—	—	6.76/11.89	2	5
$[[DV, N], N]$	N	a_3 (border)	—	—	—	—	4.24/−2.13	1	5.00/−1.21	3	9.06/11.18	37	125
$[[DV, N], N]$	R	a_1 (chunk deep)	—	—	—	—	—	—	—	—	8.40/11.66	6	25
$[DV, [N, N]]$	N	a_3 (border)	—	—	—	—	—	—	5.56/−0.15	12	5.27/7.77	86	125
$[DV, [N, N]]$	R	a_2 (chunk deepest)	—	—	—	—	—	—	—	—	5.10/7.93	5	25

Table 9
Test results for the RelClause language. The test set contains 200 sentences. The table reports the sentences that were correctly identified at each snapshot and the tree structure associated with it. When the total number does not add up to 200, it means that the other sentences were not correctly identified.

		Snapshot times				
Sentence	Tree	400	1,000	7,000	15,000	50,000
N, MV, N	$[[N, MV], N]$	4	1	0	0	1
	$[N, [MV, N]]$	0	0	4	10	67
	$[[[N, DV], N], N]$	2	0	0	0	0
	$[[N, [DV, N]], N]$	1	0	0	0	1
N, DV, N, N	$[[N, DV]], [N, N]]$	2	1	0	0	0
	$[N, [[DV, N], N]]$	1	0	1	1	16
	$[N, [DV, [N, N]]]$	0	0	0	1	46
	$[[[[[N, MV], N], R], MV], N]$	3	0	0	0	0
	$[[[[N, [MV, N]], R], MV], N]$	0	1	0	0	0
	$[[N, [[MV, N], R]], [MV, N]]$	0	0	0	0	1
	$[[N, MV], [[N, R], [MV, N]]]$	0	0	0	0	2
N, MV, N, R, MV, N	$[[N, [MV, [N, R]]], [MV, N]]$	0	0	0	0	1
	$[N, [[MV, [N, R]], [MV, N]]]$	0	0	0	0	10
	$[N, [[MV, N], [R, [MV, N]]]]$	0	0	0	0	4
	$[N, [MV, [[N, R], [MV, N]]]]$	0	0	0	0	9
	$[N, [[[MV, N], R], [MV, N]]]$	0	0	0	0	5
	$[N, [MV, [[[N, R], MV], N]]]$	0	0	0	0	2
	$[[[[N, [[DV, N], N]], R], MV], N]$	0	0	0	1	0
	$[[[N, [DV, [N, N]]], R], [MV, N]]$	0	0	0	0	1
	$[[N, [DV, [N, N]]], [R, [MV, N]]]$	0	0	0	0	6
	$[[N, [DV, [N, [N, R]]]], [MV, N]]$	0	0	0	0	1
	$[N, [[DV, [N, N]], [R, [MV, N]]]]$	0	0	0	0	5
	$[N, [[DV, [N, [N, R]]], [MV, N]]]$	0	0	0	0	2
	$[N, [[[DV, N], [N, [R, MV]]], N]]$	0	0	0	0	1
N, DV, N, N, R, MV, N	$[N, [DV, [[N, [N, R]], [MV, N]]]]$	0	0	0	0	5
	$[N, [[[DV, N], [N, R]], [MV, N]]]$	0	0	0	0	2
	$[N, [[DV, N], [[N, [R, MV]], N]]]$	0	0	0	0	1
	$[N, [[DV, N], [N, [R, [MV, N]]]]]$	0	0	0	0	2
	$[N, [[DV, N], [[N, R], [MV, N]]]]$	0	0	0	0	1
	$[N, [[[DV, N], N], [R, [MV, N]]]]$	0	0	0	0	2
	$[N, [[DV, [N, [[N, R], MV]]], N]]$	0	0	0	0	1
	$[N, [DV, [N, [[N, R], [MV, N]]]]]$	0	0	0	0	1
	$[N, [DV, [N, [N, [R, [MV, N]]]]]]$	0	0	0	0	1
	Total	13	3	5	13	197

At the end of the learning process, we observe that tree structures for different sentences share many properties. Almost all monotransitives are analyzed $[N, [MV, N]]$, most of the ditransitives are analyzed $[N, [DV, [N, N]]]$, and almost all relative clauses end with a $[MV, N]$ chunk. As stated in Section 3.1.2, a sentence of length n has C_{n-1}

possible binary tree structures. For sentences of length 3, 4, 6, and 7, we get $C_2 = 2$, $C_3 = 5$, $C_5 = 14$, and $C_6 = 42$, tree structures, respectively. Counting only structures with more than 1 instance in the last column of Table 9, we find that monotransitive sentences are associated with 1 tree structure, ditransitive sentences with 2 out of 5 possible tree structures. For relative sentences, we get 6 out of 14 and 7 out of 42, respectively. This illustrates the efficient reuse of recurring structures. The two most common tree structures associated with relative clauses of length 6 are $[N, [[MV, [N, R]], [MV, N]]]$ and $[N, [MV, [[N, R], [MV, N]]]]$, which only differ in one decision and lead to structures that are equally efficient for the task. For sentences of length 7, the length of the test set is likely too short to gather sufficient statistics, but we see that 3 structures occur more often than the others. These structures are $[[N, [DV, [N, N]]], [R, [MV, N]]]$, $[N, [[DV, [N, N]], [R, [MV, N]]]]$, and $[N, [DV, [[N, [N, R]], [MV, N]]]]$, which all analyze the ditransitive part in the same way and only differ in how the relative clause is integrated into the structure. However, the final chunk $[MV, N]$ used to predict sentence transitions is preserved.

4.2.3 Complexifying NPs. In this section, instead of adding relative clauses to the noun phrase (NP), we add and vary several shorter adnominal elements to see how the model handles these more complex and variable NPs. We add determiners (D) and one or two adjectives (A) before the noun, and prepositional phrases consisting of a preposition (P) and a noun after the noun. The language we use is specified by:

$$\begin{aligned}
\mathcal{M} &= \{S, NP, VP, N, MV, DV, A, D, P\},\\
\mathcal{T} &= \{n_1, n_2, \ldots, n_{K_n}\} \cup \{mv_1, mv_2, \ldots, mv_{K_m}\} \cup \{dv_1, dv_2, \ldots, dv_{K_v}\} \cup \{a_1, a_2, \ldots, a_{K_r}\}\\
&\quad \cup \{a_1, a_2, \ldots, a_{K_a}\} \cup \{d_1, d_2, \ldots, d_{K_d}\} \cup \{p_1, p_2, \ldots, p_{K_p}\},\\
\mathcal{S} &= S,
\end{aligned}$$

where, K_n is the total number of nouns, K_m is the total number of monotransitive verbs, K_v is the total number of ditransitive verbs, K_a is the number of adjectives, K_d is the number of determiners, and K_p is the number of prepositions. The production rules are

$$\mathcal{R} = \left\{ \begin{array}{r}
S \rightarrow NP, VP \\
NP \rightarrow N|\, D, N|\, D, A, N|\, D, A, A, N|\, N, P, N \\
VP \rightarrow MV, NP|\, DV, NP, NP \\
N \rightarrow n_1|\, n_2|\, n_3| \ldots \\
MV \rightarrow mv_1|\, mv_2|\, mv_3| \ldots \\
DV \rightarrow dv_1|\, dv_2|\, dv_3| \ldots \\
A \rightarrow a_1|\, a_2|\, a_3| \ldots \\
D \rightarrow d_1|\, d_2|\, d_3| \ldots \\
P \rightarrow p_1|\, p_2|\, p_3| \ldots
\end{array} \right\}, \qquad (6)$$

where all productions rules are equiprobable except for NPs which rewrite according to the following probabilities:

- $NP \rightarrow N$: 0.25
- $NP \rightarrow D, N$: 0.25
- $NP \rightarrow D, A, N$: 0.1875
- $NP \rightarrow D, A, A, N$: 0.0625
- $NP \rightarrow N, P, N$: 0.25

This means that double adjectives are less common than single adjective NPs, and NP with adjectives are as common as other NP production rules. We refer to this language as a ComplexNP language.

Here, we report results for a ComplexNP language with $K_n = K_m = K_v = K_a = K_d = K_p = 1$. This language contains 150 different sentence structures, which is much more than the 4 possibilities of the RelClause language above. This is why we restrict the analysis to a smaller vocabulary. Figure 5 displays the learning curve for that language. We again used the Rescorla-Wagner Q-learning algorithm with continuous border condition. As we can see, the learner successfully learns this language but as in the RelClause language, the learning curve does not follow a standard S-shaped curve. The fit to the logistic is thus imperfect to capture the learning time. We chose not to display the breakdown of the learning curve by sentence length because many different sentence structures are of the same length, for example, compare the sentences D, N, MV, N, to N, DV, N, N. Both have length 4, and there are other structures of length 4, such as N, MV, D, N. This ambiguity is even stronger for longer sentences, which justifies only displaying the overall learning curve.

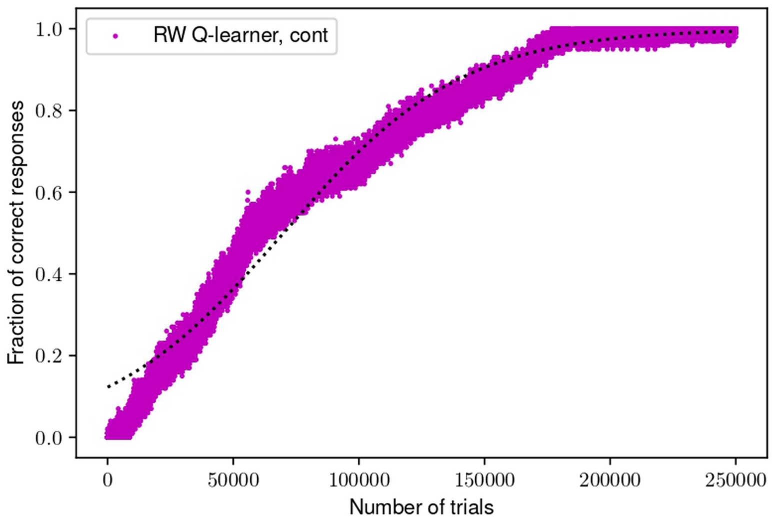


Figure 5
Learning curve for the Rescorla-Wagner Q-learning with continuous border condition for the ComplexNP language with $K_n = K_m = K_v = K_a = K_d = K_p = 1$. Fractions are obtained over 100 agents.

Table 10
Sample of tree structures of sentences from the ComplexNP language. We report only the tree structures of sentences starting by D, A, N, DV, D, A, N. The second column shows the possible tree structures and the last column report frequencies between the 500,000 and the 1,000,000 trials.

Sentence	Tree structure	Frequency
D, A, N, DV, D, A, N, N	$[[D, A], [[N, [DV, [[D, A], N]]], N]]$	0.62
	$[[D, A], [[N, [DV, [D, [A, N]]]], N]]$	0.17
	$[[[D, A], [N, [[DV, D], [A, N]]]], N]$	0.17
	$[[D, A], [N, [[DV, D], [[A, N], N]]]]$	0.04
$D, A, N, DV, D, A, N, D, N$	$[[D, A], [N, [[[DV, [[D, A], N]], D], N]]]$	0.55
	$[[[D, A], [N, [[DV, [D, [A, N]]], D]]], N]$	0.16
	$[[D, A], [N, [[[DV, D], [A, N]], [D, N]]]]$	0.10
	$[[D, A], [N, [[DV, [[D, A], N]], [D, N]]]]$	0.06
	$[[D, A], [[N, [[[DV, D], [A, N]], D]], N]]$	0.05
	$[[D, A], [N, [[DV, D], [[A, N], [D, N]]]]]$	0.03
	$[[D, A], [[N, [DV, [[D, A], [N, D]]]], N]]$	0.02
	$[[D, A], [N, [[[DV, D], [[A, N], D]], N]]]$	0.02
	$[[D, A], [N, [[DV, [D, [[A, N], D]]], N]]]$	0.01
	$[[D, A], [N, [[DV, D], [A, [[N, D], N]]]]]$	0.01
$D, A, N, DV, D, A, N, N, P, N$	$[[D, A], [[N, [DV, [[D, A], N]]], [[N, P], N]]]$	0.49
	$[[[D, A], [N, [[DV, D], [A, N]]]], [[N, P], N]]$	0.18
	$[[D, A], [[N, [DV, [D, [A, N]]]], [[N, P], N]]]$	0.15
	$[[D, A], [[[N, [DV, [[D, A], N]]], [N, P]], N]]$	0.08
	$[[D, A], [[N, [DV, [[D, A], N]]], [N, [P, N]]]]$	0.04
	$[[D, A], [[N, [[DV, D], [[A, N], N]]], [P, N]]]$	0.03
	$[[D, A], [[N, [DV, [D, [A, N]]]], [N, [P, N]]]]$	0.02
	$[[D, A], [N, [[DV, D], [[A, N], [[N, P], N]]]]]$	0.01
$D, A, N, DV, D, A, N, D, A, N$	$[[D, A], [[N, [[[DV, [[D, A], N]], D], A]], N]]$	0.56
	$[[D, A], [N, [[[[DV, [D, [A, N]]], D], A], N]]]$	0.09
	$[[D, A], [N, [[[[DV, D], [A, N]], D], [A, N]]]]$	0.09
	$[[D, A], [N, [[[DV, [D, [A, N]]], D], [A, N]]]]$	0.07
	$[[D, A], [N, [[[DV, [[D, A], N]], [D, A]], N]]]$	0.05
	$[[D, A], [N, [[[DV, D], [A, N]], [[D, A], N]]]]$	0.05
	$[[D, A], [[N, [[DV, D], [[A, N], D]]], [A, N]]]$	0.04
	$[[[D, A], [N, [DV, [[D, A], [N, D]]]]], [A, N]]$	0.02
	$[[D, A], [N, [[DV, [[D, A], N]], [[D, A], N]]]]$	0.01
	$[[D, A], [N, [[DV, [D, [A, N]]], [[D, A], N]]]]$	0.01
	$[[D, A], [N, [[[DV, D], [A, N]], [D, [A, N]]]]]$	0.01
$D, A, N, DV, D, A, N, D, A, A, N$	$[[D, A], [[N, [[[DV, [[D, A], N]], D], [A, A]]], N]]$	0.45
	$[[D, A], [N, [[[DV, [D, [A, N]]], D], [[A, A], N]]]]$	0.15
	$[[D, A], [N, [[[[DV, [[D, A], N]], D], A], [A, N]]]]$	0.09
	$[[D, A], [[N, [[DV, [[D, A], N]], [[D, A], A]]], N]]$	0.06
	$[[D, A], [N, [[[DV, D], [A, N]], [[[D, A], A], N]]]]$	0.06
	$[[D, A], [N, [[[[DV, D], [A, N]], D], [[A, A], N]]]]$	0.06
	$[[D, A], [[N, [[DV, D], [[A, N], D]]], [[A, A], N]]]$	0.04
	$[[D, A], [N, [[[DV, [D, [A, N]]], D], [A, [A, N]]]]]$	0.03
	$[[[D, A], [N, [DV, [[D, A], [N, D]]]]], [[A, A], N]]$	0.02
	$[[D, A], [N, [[DV, [D, [A, N]]], [[D, A], [A, N]]]]]$	0.01

Figure 5 is characterized by an irregular learning curve, where short plateaus can be seen. The learning curve displays a period of quick learning followed by a period of slow learning. Why this pattern occurs is unclear, but to get an idea of how the sentences are processed, we study one specific learner and record the sentences correctly identified between trial 500,000 and trial 1,000,000, i.e., after learning is complete. This provides an alternative way to test performances and is chosen for this language due to the high number of possible sentences and corresponding tree structures. Sampling the behavior after learning is complete provides better statistics how sentences are actually analyzed. Due to the large number of possible sentences, we only study 5 sentence structures, those starting with D, A, N, DV, D, A, N, that is, ditransitive sentences in which the first two NPs rewrite as D, A, N. Data for these 5 sentences as well as other structures is available in the Excel spreadsheet sentencesCompNP_paper.xlsx in the GitHub repository `https://github.com/michaudj/LanguageLearner/`. For these five sentence structures, we report the frequency of each tree structure in Table 10. Looking at the most frequent structures, we observe that they are very similar. Compare the most likely structure for the first and third sentences in the table. The structures are $[[D, A], [[N, [DV, [[D, A], N]]], N]]$ and $[[D, A], [[N, [DV, [[D, A], N]]], [[N, P], N]]]$, which only differ in the tree structure of the last NP where N is replaced by $[[N, P], N]$. Similar analysis holds for the three other sentences, which shows the high reuse of extracted information.

Another thing we can discuss is the parsimony of the grammatical information extracted. Although there are many different structures available in this table, the number of possible tree structures is actually considerably larger. It is given by the Catalan number $C_{n-1} = \frac{1}{n}\binom{2(n-1)}{n-1}$ for a sentence of length n. This means that for a sentence of length 8 such as the first sentence of Table 10 there are $C_7 = 429$ different tree structures, which is much larger than the 4 listed above. The same argument holds for longer sentences. We have about 10 tree structures for the other sentences, but the Catalan number increases extremely rapidly. In fact, we have $C_8 = 1{,}430$, $C_9 = 4{,}862$, $C_{10} = 16{,}796$, which means that the 10 tree structures for the last sentence are among the 16,796 possibilities.

5. Discussion

Our results show that our model is able to identify and reuse chunks of information that facilitates language learning in artificial languages exposing different degrees of complexity. It thus accounts, in a simplified manner, for the emergence of grammar during learning in a cognitive architecture with a short and flexible sequence memory as its core feature. The results thus lend support to the sequence hypothesis, which suggests that faithful sequence representation is a key to understanding the human language capacity.

We also find support for the hypothesis that grammar is an emergent solution to combinatorial challenges during language learning. For a simple language only containing the sentence structure noun-verb-noun, we see that the model is able to efficiently avoid the combinatorial explosion that comes with recognizing each sentence individually by only relying on noun-noun transitions to identify sentence borders. In a slightly extended language with both mono- and ditransitive verbs, containing the sentence structures noun-verb-noun and noun-verb-noun-noun, the noun-noun transition is no longer a reliable cue to sentence borders. The model then learns to use the information in the ditransitive verbs to avoid border placement between the nouns following them, and keeps the noun-noun transition as support for border placement in all other cases.

The model is also able to learn languages with extended complexity, including relative clauses and noun phrases with determiners, adjectives, and prepositional phrases. In these languages, that include common features of natural languages, we see how the model gradually learns to identify and reuse sequences of words that often occur together, like verbs and following nouns, or elements of noun phrases, thus reproducing empirical observations of natural language acquisition, where structures are initially acquired by learning frequency-based multi-word sequential chunks (Tomasello 2005; Tomasello, Kuhn, and Siegler 2008; Bybee 2002b; Arnon and Clark 2011).

The reuse of information depends on, and results in, high parsimony in the emergent tree-structures, where only a tiny fraction of all the possible tree structures actually occur. Emergent tree structures and their contained chunks seem to be optimized for the given task of sentence identification. Relying purely on transitions between words is the most economic strategy, and this strategy is also preferred by the model when word transitions contain enough information to identify sentences. When transitions are not sufficient, instead of using the all information in the full sentence strings, the model finds shorter chunks that contain enough information to make correct decisions. In a previous pilot study of this model, we show that flexible chunking that allows for all possible tree structures makes learning more efficient than a linear version of the model, which always considers the complete string it has been exposed to since its latest border placement (Jon-And and Michaud 2020). Our model thus chooses tree structures that promote compromises between economy and information, a trade-off that has been demonstrated as significant for the emergence of linguistic structure in studies of other aspects of language evolution (Kirby et al. 2015).

Studying learning curves and snapshots from our models' acquisition of different languages also shows that shorter sentences are acquired earlier than longer sentences, similarly to natural language acquisition (Hoff and Shatz 2009). Furthermore, in a language with relative clauses we see an interesting development where the model initially learns to rely on some frequent noun-verb and noun-noun transitions for identifying sentences, and then over-uses this knowledge before it has identified longer less frequent constructions or the transition to relative clauses. In order to learn to identify longer sentences, the model needs to unlearn these efficient strategies and passes through a period of very low performance before reaching a higher proficiency, where it is able to adapt strategies to different sentence structures. The model thus reproduces a U-shaped curve that can be compared to those resulting from over-regularization that typically occur in early phases of language acquisition (Bowerman 1982; Plunkett and Marchman 2020; Marcus et al. 1992; Ramscar, Dye, and McCauley 2013).

Another interesting result concerns the use of the blocking mechanism inspired by the Rescorla-Wagner model when learning to respond to a hierarchical structure. It is well observed that when animals learn to respond to compound stimuli containing one known element that is sufficient for making a good decision, they do not update any stimulus-response associations for the other elements in the compound, thus blocking unnecessary learning about new stimuli (Rescorla and Wagner 1972). Here we apply the same principle when our model is exposed to a hierarchical structure in which one state can contain many dependent sub-states, with possibilities to recruit information from all levels. A state with one or more sub-states can be compared to a compound stimulus. We tested the model without blocking, i.e., reinforcing the state-action values for all levels in the hierarchy separately and with blocking, where all levels are reinforced jointly. We found that blocking leads to a more parsimonious grammar and seems to be a more efficient strategy for extracting relevant information from linguistic input, since the more informative levels are allowed to block the reinforcement of less informative levels.

Our cognitive architecture uses the framework of Markov Decision Processes. This framework can be related to memory systems and other cognitive theories. We can interpret the state-action values as representing the long-term memory of the learners. Information about action selection during an episode of the MDP (Sutton and Barto 2018) constitutes the short-term memory. States, sub-states, and the selected actions are stored and manipulated during the processing of a sentence and this information is used for decision-making and reinforcement. This processing system resembles what Baddeley et al. call the Central Executive in their model of the working memory (Baddeley, Eysenck, and Anderson 2015). Furthermore, the limited access to information in the working memory is compatible with the chunk-and-pass principle of Christiansen and Chater (2016b) and mirrors the memory limitations discussed in Cowan (2001) and Miller (1956). All these aspects of our model stress its cognitive plausibility and highlight its minimalist nature.

The architecture we present and its learning conditions differ from those of other grammar induction models in some aspects. Our model is simple and transparent, lacks assumptions on pre-defined categories, and uses cognitively plausible localized decision-making and learning. Learning is based purely on incremental exposure to a raw unsegmented stream of words, where sentence borders are used as cues for reinforcement. The model succeeds in identifying sentences and finds parsimonious tree structures to support this task, which may serve as a basis for more extensive generalizations. This indicates that the more elaborate predispositions or more extensive data processing involved in other models may not be necessary for grammar induction. Moreover, our model provides a starting point for operationalization of usage-based language learning (Bybee 2006; Ellis, O'Donnell, and Römer 2015; Tomasello 2003), and the results indicate that the architecture and task offers a promising direction for developing models that can provide a more complete insight into how humans induce grammar from data.

Our minimal temporal difference model is also different from connectionist models, including LLMs, as it aims for maximal simplicity. This makes results easier to trace and interpret, and it also enables us to make as few assumptions as possible and see how far they can take us in understanding the human language learning process. Our results indicate that limiting working memory and training data may be cues to understanding human language learning, as our model learns to find the shortest and most informative chunks as a strategy to overcome these constraints. Our results also indicate that a limited sequence memory combined with flexible chunking are central components for learning language.

The model we explore is limited by the lack of generalization and abstraction, preventing it from scaling to larger languages. We see that learning times increase when increasing the size of the vocabulary or the complexity of the artificial languages. While our results provide an initial proof of concept that the minimal cognitive architecture we present is able to extract grammatical information, future studies should explore generalization to generate abstract categories that would enable testing the model on languages with larger vocabularies and eventually apply it to natural languages. Emergent abstract categories would then be expected to resemble grammatical categories like word classes or syntactic constituents. Error-correction models have previously successfully accounted for the emergence of phones (Milin, Tucker, and Divjak 2023) and the learning of regular and irregular morphological forms (Ramscar, Dye, and McCauley 2013; MacWhinney et al. 1989) in a human-like manner from exposure to limited parts of language that are relevant to these tasks. The modeling of the emergence of syntactic categories is arguably a more challenging and more interesting problem, as it needs to involve tracing relationships between words and constituents at different levels, and requires exposure to a somewhat complete language.

A generalization system in our model should generalize over the information that the model uses, namely, information about sequences and chunks. A possible approach to such generalization is the mathematical theory of type systems, being the simplest mathematical framework for handling ordering, composition, and abstraction of things (Heunen, Sadrzadeh, and Grefenstette 2013), thus incorporating the features sequence memory, chunking, and schematizing. In generative linguistics, an application of this framework is made in the theory of syntactic types (Lambek 1958). An advantage of syntactic types in comparison to other grammatical formalisms is that a formula encoding how a unit can be combined with other units is associated with each element, which can be a word or a chunk. This allows for local support of decisions on whether and how to chunk elements in incremental processing, and the learner does not need to have access to an entire grammatical system at the moment of a decision, but only to information that concerns the units that are being processed. This makes the framework suitable for implementation in a reinforcement learning paradigm. In a pilot study, we have modified the theory of syntactic types to be dynamic and compatible with usage-based learning. In this novel framework, which we call an *evolutionary type system*, no assumptions are made on primitives or types initially, other than that there are meaningful units (sentences). The lack of predefined categories or tags, and the open-ended dynamics of this system set it apart from other categorial-based formalisms (Steedman and Baldridge 2011; Kogkalidis, Moortgat, and Moot 2020; Bisk and Hockenmaier 2012). The evolutionary type system allows the learner to make the simplest possible generalization based on sequential order and chunking when a sentence is correctly identified, which is that each element in the chunk can become a meaningful unit, if combined with the preceding and/or following element(s) in this chunk. Initially the system starts with a single primitive representing sentences and corresponding to the start symbol of context free grammars. New primitives and types are then invented when needed, replicated through type assignment and selected based on their usefulness for sentence segmentation. The innovation and selection processes positions this system in the general category of evolutionary systems. Our pilot results show that under these conditions, functional abstract categories can emerge for short sentences (Jon-And and Michaud 2024), indicating that this is a promising framework for further development.

Another limitation of our model is that it only considers syntax and not semantics. This complicates the learning task since the meaning of words highly constrain the possible parsing of a sentence. Form-meaning pairings are considered fundamental for usage-based learning and the compatible framework of construction grammar (Langacker 2002; Goldberg 2007; Croft and Cruse 2004; Tomasello 2005). In LLMs, even though words are not mapped onto real-world meanings, meaning is represented in a distributional sense and is central to their performance (Piantasodi and Hill 2022; Manning et al. 2020; Amaratunga 2023). This distributional, text-related sense of meaning could be implemented in a minimalist model like ours and would likely be a key for it to perform well on natural language corpora. A further possible future development of our model would be to use the emergent grammar it abstracts for linguistic production, and to let production and feedback to production be part of the learning process. This would be an important step towards testing the model's ability to account for the complete language acquisition process. There are still quite a few steps that need to be taken to achieve this, but our initial results motivate further development and exploration of this minimal cognitive architecture.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments and suggestions. We also thank our colleagues at the Centre for Cultural Evolution, Stockholm University, for inspiring and helpful discussions. This work was supported by the Swedish Research Council (VR 2022-02737).

References

Amalric, Marie, Liping Wang, Pierre Pica, Santiago Figueira, Mariano Sigman, and Stanislas Dehaene. 2017. The language of geometry: Fast comprehension of geometrical primitives and rules in human adults and preschoolers. *PLoS Computational Biology*, 13(1):e1005273. `https://doi.org/10.1371/journal.pcbi.1005273`, PubMed: 28125595

Amaratunga, Thimira. 2023. *Understanding Large Language Models: Learning their Underlying Concepts and Technologies*. Springer. `https://doi.org/10.1007/979-8-8688-0017-7`

Arnon, Inbal and Eve V. Clark. 2011. Why brush your teeth is better than teeth—Children's word production is facilitated in familiar sentence-frames. *Language Learning and Development*, 7(2):107–129. `https://doi.org/10.1080/15475441.2010.505489`

Baddeley, Alan, Michael W. Eysenck, and Michael C. Anderson. 2015. *Memory*. Psychology Press. `https://doi.org/10.4324/9781315749860`

Baxter, Gareth J., Richard A. Blythe, William Croft, and Alan J. McKane. 2006. Utterance selection model of language change. *Physical Review E*, 73(4):046118. `https://doi.org/10.1103/PhysRevE.73.046118`, PubMed: 16711889

Beltagy, Iz, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Berant, Jonathan, Yaron Gross, Matan Mussel, Ben Sandbank, Eytan Ruppin, and Shimon Edelman. 2007. Boosting unsupervised grammar induction by splitting complex sentences on function words. In *Proceedings of the 31st Boston University Conference on Language Development*, pages 93–104.

Bisk, Yonatan and Julia Hockenmaier. 2012. Simple robust grammar induction with combinatory categorial grammars. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 1643–1649. `https://doi.org/10.1609/aaai.v26i1.8355`

Bod, Rens. 2006. An all-subtrees approach to unsupervised parsing. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 865–872. `https://doi.org/10.3115/1220175.1220284`

Bod, Rens. 2009. From exemplar to grammar: A probabilistic analogy-based model of language learning. *Cognitive Science*, 33(5):752–793. `https://doi.org/10.1111/j.1551-6709.2009.01031.x`, PubMed: 21585486

Bouton, M. E. 2016. *Learning and Behavior: A Contemporary Synthesis*, 2nd edition. Sinauer.

Bowerman, Melissa. 1982. Starting to talk worse: Clues to language acquisition from children's late speech errors. In *U shaped Behavioral Growth*. Academic Press, pages 101–145. `https://doi.org/10.1016/B978-0-12-673020-3.50012-4`

Brodsky, Peter and Heidi Waterfall. 2007. Characterizing motherese: On the computational structure of child-directed language. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 29.

Brown, Tom, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Bybee, Joan. 2002a. Phonological evidence for exemplar storage of multiword sequences. *Studies in Second Language Acquisition*, 24(2):215–221. `https://doi.org/10.1017/S0272263102002061`

Bybee, Joan. 2002b. Sequentiality as the basis of constituent structure. *The Evolution of Language Out of Pre-language*, 53:109–134. `https://doi.org/10.1075/tsl.53.07byb`

Bybee, Joan. 2006. From usage to grammar: The mind's response to repetition. *Language*, pages 711–733. `https://doi.org/10.1353/lan.2006.0186`

Bybee, Joan L. 1985. *Morphology: A Study of the Relation between Meaning and Form*. John Benjamins Publishing. `https://doi.org/10.1075/tsl.9`

Chomsky, N. 1957. *Syntactic Structures*. Mouton, The Hague/Paris. `https://doi.org/10.1515/9783112316009`

Chomsky, Noam. 1959. On certain formal properties of grammars. *Information and*

Control, 2(2):137–167. https://doi.org/10.1016/S0019-9958(59)90362-6

Chomsky, Noam. 2002. *Syntactic Structures*. Mouton de Gruyter. https://doi.org/10.1515/9783110218329

Christiansen, Morten H. and Inbal Arnon. 2017. More than words: The role of multiword sequences in language learning and use. *Topics in Cognitive Science*, 9(3):542–551. https://doi.org/10.1111/tops.12274, PubMed: 28503906

Christiansen, Morten H. and Nick Chater. 2001. *Connectionist Psycholinguistics*. Greenwood Publishing Group.

Christiansen, Morten H. and Nick Chater. 2016a. *Creating Language: Integrating Evolution, Acquisition, and Processing*. MIT Press. https://doi.org/10.7551/mitpress/10406.001.0001

Christiansen, Morten H. and Nick Chater. 2016b. The now-or-never bottleneck: A fundamental constraint on language. *Behavioral and Brain Sciences*, 39. https://doi.org/10.1017/S0140525X1500031X, PubMed: 25869618

Christiansen, Morten H. and Simon Kirby. 2003. Language evolution: Consensus and controversies. *Trends in Cognitive Sciences*, 7(7):300–307. https://doi.org/10.1093/acprof:oso/9780199244843.001.0001, PubMed: 12860188

Christiansen, Morten H. and Maryellen C. MacDonald. 2009. A usage-based approach to recursion in sentence processing. *Language Learning*, 59:126–161. https://doi.org/10.1111/j.1467-9922.2009.00538.x

Cornish, Hannah, Rick Dale, Simon Kirby, and Morten H. Christiansen. 2017. Sequence memory constraints give rise to language-like structure through iterated learning. *PloS ONE*, 12(1):e0168532. https://doi.org/10.1371/journal.pone.0168532, PubMed: 28118370

Cowan, Nelson. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114. https://doi.org/10.1017/S0140525X01003922, PubMed: 11515286

Croft, William. 2001. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press on Demand. https://doi.org/10.1093/acprof:oso/9780198299554.001.0001

Croft, William and D. Alan Cruse. 2004. *Cognitive Linguistics*. Cambridge University Press. https://doi.org/10.1017/CBO9780511803864

Ellis, Kevin, Lucas Morales, Mathias Sablé-Meyer, Armando Solar-Lezama, and Josh Tenenbaum. 2018. Learning libraries of subroutines for neurally–guided Bayesian program induction. *Advances in Neural Information Processing Systems*, 31.

Ellis, Nick C., Matthew Brook O'Donnell, and Ute Römer. 2015. Usage-based language learning. *The Handbook of Language Emergence*, pages 163–180. https://doi.org/10.1002/9781118346136.ch7

Elman, Jeffrey L. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211. https://doi.org/10.1016/0364-0213(90)90002-E

Elman, Jeffrey L. 1996. *Rethinking Innateness: A Connectionist Perspective on Development*, volume 10. MIT Press. https://doi.org/10.7551/mitpress/5929.001.0001

Enquist, Magnus, Stefano Ghirlanda, and Johan Lind. 2023. *The Human Evolutionary Transition: From Animal Intelligence to Culture*. Princeton University Press. https://doi.org/10.23943/princeton/9780691240770.001.0001

Enquist, Magnus, Johan Lind, and Stefano Ghirlanda. 2016. The power of associative learning and the ontogeny of optimal behaviour. *Royal Society Open Science*, 3(11):160734. https://doi.org/10.1098/rsos.160734, PubMed: 28018662

Fodor, Jerry A. 1983. *The Modularity of Mind*. MIT Press. https://doi.org/10.7551/mitpress/4737.001.0001

Frank, Stefan L., Rens Bod, and Morten H. Christiansen. 2012. How hierarchical is language use? *Proceedings of the Royal Society B: Biological Sciences*, 279(1747):4522–4531. https://doi.org/10.1098/rspb.2012.1741, PubMed: 22977157

Ghirlanda, Stefano, Johan Lind, and Magnus Enquist. 2017. Memory for stimulus sequences: A divide between humans and other animals? *Open Science*, 4(6):161011. https://doi.org/10.1098/rsos.161011, PubMed: 28680660

Goldberg, A. E. 2007. *Constructions at Work: The Nature of Generalization in Language*. Oxford linguistics. Oxford University Press.

Griffiths, Thomas L., Nick Chater, Charles Kemp, Amy Perfors, and Joshua B. Tenenbaum. 2010. Probabilistic models of cognition: Exploring representations and inductive biases. *Trends in Cognitive*

Sciences, 14(8):357–364. https://doi.org/10.1016/j.tics.2010.05.004, PubMed: 20576465

Haselgrove, Mark. 2016. Overcoming associative learning. *Journal of Comparative Psychology*, 130(3):226–240. https://doi.org/10.1037/a0040180, PubMed: 26986019

Heunen, Chris, Mehrnoosh Sadrzadeh, and Edward Grefenstette. 2013. *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*. Oxford University Press, USA. https://doi.org/10.1093/acprof:oso/9780199646296.001.0001

Heyes, Cecilia. 2012a. Simple minds: A qualified defence of associative learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1603):2695–2703. https://doi.org/10.1098/rstb.2012.0217, PubMed: 22927568

Heyes, Cecilia. 2012b. What's social about social learning? *Journal of Comparative Psychology*, 126(2):193–202. https://doi.org/10.1037/a0025180, PubMed: 21895355

Heyes, Cecilia. 2018. *Cognitive Gadgets: The Cultural Evolution of Thinking*. Harvard University Press. https://doi.org/10.2307/j.ctv24trbqx, https://doi.org/10.4159/9780674985155

Hochmann, Jean Rémy, Mahan Azadpour, and Jacques Mehler. 2008. Do humans really learn AnBn artificial grammars from exemplars? *Cognitive Science*, 32(6):1021–1036. https://doi.org/10.1080/03640210801897849

Hoff, Erika and Marilyn Shatz. 2009. *Blackwell Handbook of Language Development*. John Wiley & Sons.

Jon-And, A. and J. Michaud. 2024. Emergent grammar from a minimal cognitive architecture. In *The Evolution of Language: Proceedings of the 15th International Conference (Evolang XV)*.

Jon-And, Anna, Markus Jonsson, Johan Lind, Stefano Ghirlanda, and Magnus Enquist. 2023. Sequence representation as an early step in the evolution of language. *PLOS Computational Biology*, 19(12):e1011702. https://doi.org/10.1371/journal.pcbi.1011702, PubMed: 38091352

Jon-And, Anna and Jérôme Michaud. 2020. Minimal prerequisites for processing language structure: A model based on chunking and sequence memory. In *EvoLang XIII*, pages 200–208.

Kirby, S., H. Cornish, and K. Smith. 2008. Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, 105(31):10681–10686. https://doi.org/10.1073/pnas.0707835105, PubMed: 18667697

Kirby, Simon, Monica Tamariz, Hannah Cornish, and Kenny Smith. 2015. Compression and communication in the cultural evolution of linguistic structure. *Cognition*, 141:87–102. https://doi.org/10.1016/j.cognition.2015.03.016, PubMed: 25966840

Kogkalidis, Konstantinos, Michael Moortgat, and Richard Moot. 2020. Neural proof nets. *arXiv preprint arXiv:2009.12702*. https://doi.org/10.18653/v1/2020.conll-1.3

Kolodny, Oren and Shimon Edelman. 2018. The evolution of the capacity for language: The ecological context and adaptive value of a process of cognitive hijacking. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 373(1743):20170052. https://doi.org/10.1098/rstb.2017.0052, PubMed: 29440518

Kolodny, Oren, Shimon Edelman, and Arnon Lotem. 2015. Evolution of protolinguistic abilities as a by-product of learning to forage in structured environments. *Proceedings of the Royal Society B: Biological Sciences*, 282(1811):20150353. https://doi.org/10.1098/rspb.2015.0353, PubMed: 26156764

Kuhl, Patricia K. 2004. Early language acquisition: Cracking the speech code. *Nature Reviews Neuroscience*, 5(11):831–843. https://doi.org/10.1038/nrn1533, PubMed: 15496861

Lambek, Joachim. 1958. The mathematics of sentence structure. *The American Mathematical Monthly*, 65(3):154–170. https://doi.org/10.1080/00029890.1958.11989160

Langacker, Ronald W. 1987. *Foundations of Cognitive Grammar: Volume I: Theoretical Prerequisites*, volume 1. Stanford University Press.

Langacker, Ronald W. 2002. *Concept, Image, and Symbol*. Walter de Gruyter Inc. https://doi.org/10.1515/9783110857733

Liang, Xinnian, Bing Wang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. 2023. Unleashing infinite-length input capacity for large-scale language models with self-controlled memory system. *arXiv preprint arXiv:2304.13343*.

Lind, J., S. Ghirlanda, and M. Enquist. 2022. Evolution of memory systems in animals.

In M. Krause, K. L. Hollis, and M. R. Papini, editors, *Evolution of Learning and Memory Mechanisms*. Cambridge University Press, pages 339–358. https://doi.org/10.1017/9781108768450.023

Lind, Johan. 2018. What can associative learning do for planning? *Royal Society Open Science*, 5(11):180778. https://doi.org/10.1098/rsos.180778, PubMed: 30564390

Lind, Johan, Vera Vinken, Markus Jonsson, Stefano Ghirlanda, and Magnus Enquist. 2023. A test of memory for stimulus sequences in great apes. *PLos ONE*, 18(9):e0290546. https://doi.org/10.1371/journal.pone.0290546, PubMed: 37672549

MacDonald, Suzanne E. 1993. Delayed matching-to-successive-samples in pigeons: Short-term memory for item and order information. *Animal Learning & Behavior*, 21(1):59–67. https://doi.org/10.3758/BF03197977

Mackintosh, Nicholas John. 1983. *Conditioning and Associative Learning*. Oxford University Press, USA.

MacWhinney, Brian, Jared Leinbach, Roman Taraban, and Janet McDonald. 1989. Language learning: Cues or rules? *Journal of Memory and Language*, 28(3):255–277. https://doi.org/10.1016/0749-596X(89)90033-8

Manning, Christopher and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Manning, Christopher D., Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054. https://doi.org/10.1073/pnas.1907367117, PubMed: 32493748

Marcus, Gary F., Steven Pinker, Michael Ullman, Michelle Hollander, T. John Rosen, Fei Xu, and Harald Clahsen. 1992. Overregularization in language acquisition. *Monographs of the Society for Research in Child Development*, 57(4):181–182. https://doi.org/10.2307/1166115, PubMed: 1518508

McCauley, Stewart M. and Morten H. Christiansen. 2019. Language learning as language use: A cross-linguistic model of child language development. *Psychological Review*, 126(1):1–51. https://doi.org/10.1037/rev0000126, PubMed: 30604987

McClelland, James L., Matthew M. Botvinick, David C. Noelle, David C. Plaut, Timothy T. Rogers, Mark S. Seidenberg, and Linda B. Smith. 2010. Letting structure emerge: Connectionist and dynamical systems approaches to cognition. *Trends in Cognitive Sciences*, 14(8):348–356. https://doi.org/10.1016/j.tics.2010.06.002, PubMed: 20598626

Michaud, Jérôme. 2019. Dynamic preferences and self-actuation of changes in language dynamics. *Language Dynamics and Change*, 9(1):61–103. https://doi.org/10.1163/22105832-00901003

Milin, Petar, Benjamin V. Tucker, and Dagmar Divjak. 2023. A learning perspective on the emergence of abstractions: The curious case of phone(me)s. *Language and Cognition*, pages 1–23. https://doi.org/10.1017/langcog.2023.11

Miller, George A. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97. https://doi.org/10.1037/h0043158, PubMed: 13310704

Muralidaran, Vigneshwaran, Irena Spasić, and Dawn Knight. 2021. A systematic review of unsupervised approaches to grammar induction. *Natural Language Engineering*, 27(6):647–689. https://doi.org/10.1017/S1351324920000327

Newell, Allen. 1994. *Unified Theories of Cognition*. Harvard University Press.

Nivre, Joakim, Marie-Catherine De Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal dependencies v2: An evergrowing multilingual treebank collection. *arXiv preprint arXiv:2004.10643*.

Nowak, Martin A., Natalia L. Komarova, and Partha Niyogi. 2002. Computational and evolutionary aspects of language. *Nature*, 417(6889):611–617. https://doi.org/10.1038/nature00771, PubMed: 12050656

Pavlov, Ivan P. 1949. Conditioned responses. *Readings in General Psychology*, pages 249–267. https://doi.org/10.1037/11352-036

Peters, Ann M. 2013. Language segmentation: Operating principles for the perception and analysis of language. In *The Crosslinguistic Study of Language Acquisition*. Psychology Press, pages 1029–1067.

Piantadosi, S. T. 2023. Modern language models refute Chomsky's approach to language. *Lingbuzz Preprint, lingbuzz*, 7180.

Piantadosi, Steven T., Joshua B. Tenenbaum, and Noah D. Goodman. 2016. The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, 123(4):392–424. https://doi.org/10.1037/a0039980, PubMed: 27077241

Piantasodi, Steven T. and Felix Hill. 2022. Meaning without reference in large language models. *arXiv preprint arXiv:2208.02957*.

Pinker, Steven and Ray Jackendoff. 2005. The faculty of language: What's special about it? *Cognition*, 95(2):201–236. https://doi.org/10.1016/j.cognition.2004.08.004, PubMed: 15694646

Planton, Samuel, Timo van Kerkoerle, Leïla Abbih, Maxime Maheu, Florent Meyniel, Mariano Sigman, Liping Wang, Santiago Figueira, Sergio Romano, and Stanislas Dehaene. 2021. A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans. *PLoS Computational Biology*, 17(1):e1008598. https://doi.org/10.1371/journal.pcbi.1008598, PubMed: 33465081

Plunkett, Kim and Virginia Marchman. 2020. U-shaped learning and frequency effects in a multilayered perceptron: Implications for child language acquisition. In R. Ellis & G. W. Humphreys, editors, *Connectionist Psychology: A Textbook with Readings*. Psychology Press, pages 487–526. https://doi.org/10.4324/9781315784779-15

Post, Matt and Daniel Gildea. 2013. Bayesian tree substitution grammars as a usage-based approach. *Language and Speech*, 56(3):291–308. https://doi.org/10.1177/0023830913484901, PubMed: 24416958

Ramscar, Michael, Melody Dye, and Stewart M. McCauley. 2013. Error and expectation in language learning: The curious absence of "mouses" in adult speech. *Language*, pages 760–793. https://doi.org/10.1353/lan.2013.0068

Read, Dwight W., Héctor M. Manrique, and Michael J. Walker. 2022. On the working memory of humans and great apes: Strikingly similar or remarkably different? *Neuroscience & Biobehavioral Reviews*. https://doi.org/10.1016/j.neubiorev.2021.12.019, PubMed: 34919985

Reali, Florencia and Thomas L. Griffiths. 2009. The evolution of frequency distributions: Relating regularization to inductive biases through iterated learning. *Cognition*, 111(3):317–328. https://doi.org/10.1016/j.cognition.2009.02.012, PubMed: 19327759

Rescorla, R. A. and A. R. Wagner. 1972. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In *Classical Conditioning: Current Research and Theory*. Appleton-Century-Crofts, pages 64–69.

Roberts, W. A. and D. S. Grant. 1976. Studies of short-term memory in the pigeon using the delayed matching to sample procedure. In R. T. Davis, D. L. Medin, and W. A. Roberts, editors, *Processes of Animal Memory*. Erlbaum, pages 79–112.

Rule, Joshua, Eric Schulz, Steven T. Piantadosi, and Joshua B. Tenenbaum. 2018. Learning list concepts through program induction. *BioRxiv*, 321505. https://doi.org/10.1101/321505

Rule, Joshua S., Joshua B. Tenenbaum, and Steven T. Piantadosi. 2020. The child as hacker. *Trends in Cognitive Sciences*, 24(11):900–915. https://doi.org/10.1016/j.tics.2020.07.005, PubMed: 33012688

Saffran, Jenny R. 2001. Words in a sea of sounds: The output of infant statistical learning. *Cognition*, 81(2):149–169. https://doi.org/10.1016/S0010-0277(01)00132-9, PubMed: 11376640

Saffran, Jenny R., Richard N. Aslin, and Elissa L. Newport. 1996. Statistical learning by 8-month-old infants. *Science*, 274(5294):1926–1928. https://doi.org/10.1126/science.274.5294.1926, PubMed: 8943209

Sanders, Lisa D., Helen J. Neville, and Marty G. Woldorff. 2002. Speech segmentation by native and non-native speakers. *Journal of Speech, Language, and Hearing Research*, 45(3):519–530. https://doi.org/10.1044/1092-4388(2002/041)

Sanford, Anthony J. and Patrick Sturt. 2002. Depth of processing in language comprehension: Not noticing the evidence. *Trends in Cognitive Sciences*, 6(9):382–386. https://doi.org/10.1016/S1364-6613(02)01958-7, PubMed: 12200180

Servan-Schreiber, Emile and John R. Anderson. 1990. Learning artificial grammars with competitive chunking. *Journal of Experimental Psychology: Learning,* *Memory, and Cognition*, 16(4):592. https://doi.org/10.1037//0278-7393.16.4.592

Shain, Cory, William Bryce, Lifeng Jin, Victoria Krakovna, Finale Doshi-Velez, Timothy Miller, William Schuler, and Lane Schwartz. 2016. Memory-bounded left-corner unsupervised grammar induction on child-directed input. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 964–975.

Silver, David, Satinder Singh, Doina Precup, and Richard S. Sutton. 2021. Reward is enough. *Artificial Intelligence*, 299:103535. https://doi.org/10.1016/j.artint.2021.103535

Skinner. 1965. *Science and Human Behavior*. Simon and Schuster.

Solan, Zach, David Horn, Eytan Ruppin, and Shimon Edelman. 2003. Unsupervised context sensitive language acquisition from a large corpus. *Advances in Neural Information Processing Systems*, 16.

Steedman, Mark and Jason Baldridge. 2011. Combinatory categorial grammar. *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, pages 181–224. https://doi.org/10.1002/9781444395037.ch5

Steels, Luc and Martin Loetzsch. 2012. The grounded naming game. *Experiments in Cultural Language Evolution*, 3:41–59. https://doi.org/10.1075/ais.3.04ste

Sutton, Richard S. and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Tenenbaum, Joshua B., Charles Kemp, Thomas L. Griffiths, and Noah D. Goodman. 2011. How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022):1279–1285. https://doi.org/10.1126/science.1192788, PubMed: 21393536

Tomasello, Michael. 2003. *Constructing a Language: A Usage-based Theory of Language Acquisition*. Harvard University Press.

Tomasello, Michael. 2005. *Constructing a Language: A Usage-based Theory of Language Acquisition*. Harvard University Press. https://doi.org/10.2307/j.ctv26070v8

Tomasello, Michael, D. Kuhn, and R. Siegler. 2008. Acquiring linguistic constructions. *Child and Adolescent Development*, page 263. https://doi.org/10.1002/9780470147658.chpsy0206

Udden, Julia, Martin Ingvar, Peter Hagoort, and Karl M. Petersson. 2012. Implicit acquisition of grammars with crossed and nested non-adjacent dependencies: Investigating the push-down stack model. *Cognitive Science*, 36(6):1078–1101. https://doi.org/10.1111/j.1551-6709.2012.01235.x, PubMed: 22452530

Ullman, Tomer D., Noah D. Goodman, and Joshua B. Tenenbaum. 2012. Theory learning as stochastic search in the language of thought. *Cognitive Development*, 27(4):455–480. https://doi.org/10.1016/j.cogdev.2012.07.005

van der Velde, Frank, Jamie Forth, Deniece S. Nazareth, and Geraint A. Wiggins. 2017. Linking neural and symbolic representation and processing of conceptual structures. *Frontiers in Psychology*, 8:1297. https://doi.org/10.3389/fpsyg.2017.01297, PubMed: 28848460

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Warstadt, Alex and Samuel R. Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In *Algebraic Structures in Natural Language*. CRC Press, pages 17–60. https://doi.org/10.1201/9781003205388-2

Wasserman, Edward A., Andrew G. Kain, and Ellen M. O'Donoghue. 2023. Resolving the associative learning paradox by category learning in pigeons. *Current Biology*, 33(6):1112–1116. https://doi.org/10.1016/j.cub.2023.01.024, PubMed: 36754051

Wiggins, Geraint A. 2020. Creativity, information, and consciousness: The information dynamics of thinking. *Physics of Life Reviews*, 34:1–39. https://doi.org/10.1016/j.plrev.2018.05.001, PubMed: 29803403

Yogatama, Dani, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2016. Learning to compose words into sentences with reinforcement learning. *arXiv preprint arXiv:1611.09100*.