# Supporting Text

## 1. Algorithm Details

Consider a corpus of $m$ sentences (sequences) of variable length, each expressed in terms of a lexicon of finite size $N$. The sentences in the corpus correspond to $m$ different paths in a pseudograph (a nonsimple graph in which both loops and multiple edges are permitted) whose vertices are the unique lexicon entries, augmented by two special symbols, *begin* and *end*. Each of the N nodes has a number of incoming paths that is equal to the number of outgoing paths. Fig. 4 illustrates the type of structure that we seek, namely, the bundling of paths, signifying a relatively high probability associated with a sub-structure that can be identified as a pattern. To extract it from the data, two probability functions are defined over the graph for any given *search path* $\mathbf{S} = (e_1 \rightarrow e_2 \rightarrow \cdots \rightarrow e_k) = (e_1; e_k)$.* The first one, $P_R(e_i; e_j)$, is the right-moving ratio of fan-through flux of paths at $e_j$ to fan-in flux of paths at $e_{j-1}$, starting at $e_i$ and moving along the subpath $e_i \rightarrow e_{i+1} \rightarrow e_{i+2} \cdots \rightarrow e_{j-1}$

$$P_R(e_i; e_j) = p(e_j | e_i e_{i+1} e_{i+2} ... e_{j-1}) = \frac{l(e_i; e_j)}{l(e_i; e_{j-1})}, \quad \textbf{[1]}$$

where $l(e_i; e_j)$ is the number of occurrences of subpaths $(e_i; e_j)$ in the graph. Proceeding in the opposite direction, from the right end of the path to the left, we define the left-going probability function $P_L$

$$P_L(e_j; e_i) = p(e_i | e_{i+1} e_{i+2} ... e_{j-1} e_j) = \frac{l(e_j; e_i)}{l(e_j; e_{i+1})}, \quad \textbf{[2]}$$

and note that

$$P_R(e_i; e_i) = P_L(e_i; e_i) = \frac{l(e_i)}{\sum_{x=0}^{N} l(e_x)} \quad \textbf{[3]}$$

where N is the total number of vertices in the graph. Clearly, both functions vary between 0 and 1 and are specific to the path in question. The MEX algorithm is defined in terms of these functions and their ratios. In Fig. 5, $P_R$ first increases because some other paths join the search path to form a coherent bundle, then decreases at $e4$, because many paths leave it at $e4$. To quantify this decline of $P_R$, which we interpret as an indication of the end of the candidate pattern, we define a *decrease ratio*, $D_R(e_i; e_j)$, whose value at $e_j$ is $D_R(e_i; e_j) = P_R(e_i; e_j)/P_R(e_i; e_{j-1})$, and require that it be smaller than a preset *cutoff parameter* $\eta < 1$ [in the present example, $D_R(e_1, e_5) = P_R(e_1, e_5)/P_R(e_1, e_4) < \frac{1}{3}$].

In a similar manner, the value of $P_L$ increases leftward; the point $e_2$ at which it first shows a decrease $D_L(e_j; e_i) = P_L(e_j; e_i)/P_L(e_j; e_{i+1}) < \eta$ can be interpreted as the starting point of the candidate pattern. Large values of $D_L$ and $D_R$ signal a divergence of the paths that constitute the bundle, thus making a pattern-candidate. Since the relevant probabilities [$P_R(e_i; e_j)$ and $P_L(e_j; e_i)$] are determined by finite and possibly small numbers of paths [$l(e_i; e_j)$ out of $l(e_i; e_{j-1})$], we face the problem of small-sample statistics. We find it useful therefore to supplement conditions such as $D_R(e_i; e_j) < \eta$ by a significance test based on binomial probabilities, as follows

*In general the notation $(e_i; e_j), j > i$ corresponds to a rightward subpath of **S**, starting with $e_i$ and ending with $e_j$. A leftward subpath of **S**, starting with $e_j$ and ending with $e_i$ is denoted by $(e_j; e_i), i < j$.

$$B(e_i;e_j) = \sum_{x=0}^{l(e_i;e_j)} Binom(x, l(e_i;e_{j-1}), \eta P_R(e_i;e_{j-1})) < \alpha; \alpha \ll 1. \quad \textbf{[4]}$$

We calculate both $P_L$ and $P_R$ from all the possible starting points (such as $e1$ and $e4$ in the example of Fig. 5), traversing each path leftward and rightward, correspondingly. This defines a matrix of the form

$$M_{ij}(\mathbf{S}) = \begin{cases} P_R(e_i;e_j) & \text{if } i > j \\ P_L(e_j;e_i) & \text{if } i < j \\ P(e_i) & \text{if } i = j. \end{cases} \quad \textbf{[5]}$$

One can write $M(\mathbf{S})$ in its explicit form, namely, as an instantiation of a variable-order Markov model up to order $k$, which is the length of the search-path

$$\mathbf{M} \doteq \begin{pmatrix} p(e_1) & p(e_1|e_2) & p(e_1|e_2e_3) & \dots & p(e_1|e_2e_3\dots e_k) \\ p(e_2|e_1) & p(e_2) & p(e_2|e_3) & \dots & p(e_2|e_3e_4\dots e_k) \\ p(e_3|e_1e_2) & p(e_3|e_2) & p(e_3) & \dots & p(e_3|e_4e_5\dots e_k) \\ & \vdots & \vdots & \vdots & \vdots \\ p(e_k|e_1e_2\dots e_{k-1}) & p(e_k|e_2e_3\dots e_{k-1}) & p(e_k|e_3e_4\dots e_{k-1}) & \dots & p(e_k) \end{pmatrix}. \quad \textbf{[6]}$$

Given the matrix $\mathbf{M}(\mathbf{S})$, we identify all the significant $D_R(e_a;e_b)$ and $D_L(e_d;e_c)$ ($1 \le a,b,c,d \le k$) and their coinciding pairs ($D_R(e_a;e_b), D_L(e_c;e_d)$), requiring that $a < d < b < c$. The pair with the most significant scores [on both sides, $B(e_a;e_b)$ and $B(e_d;e_c)$] is declared as the leading pattern ($e_{d+1}$; $e_{b-1}$).

# 2. Learning a Simple Context-Free Grammar (CFG)

## 2.1. Replicating the study of Adriaans and Vervoort (2002): EMILE 4.1

We replicated one of the experiments of ref. 1 ("A 2000 Sentences Sample", p. 8). The aim of the original experiment was to reconstruct a specific CFG (29 terminals and 7 rules) from a corpus of 2,000 sentences using the EMILE 4.1 algorithm. The results of applying the ADIOS algorithm to a 2,000-sentence corpus randomly generated from the given context-free grammar are shown in Table 1. The algorithm (used in its default Mode A, $\eta = 0.6$, $\alpha = 0.01$, recursion depth set to 15) yielded 28 patterns and 9 equivalence classes and achieved $100\%$ precision and $99\%$ recall. In comparison, the EMILE algorithm, as reported in ref. 1, induced 3,000–4,000 rules (the recall/precision performance of the EMILE algorithm was not stated). Table 1 shows a comparison between the induced grammar and its target grammar. The upper part of the table contains the extracted equivalence classes and their target counterparts, demonstrating the ability of ADIOS to identify most of the target classes (except one, E43). The lower part of the table shows that ADIOS distills a set of rules that is larger than the original one (but equivalent to it).

## 2.2. Inferring the TA1 grammar: supplement to Fig. 3$A$

Tables 2 to 5 show the performance of an ADIOS model trained on extremely small corpora (200 sentences) generated by the TA1 artificial grammar (listed in Table 6). The tables present the recall-precision values (with their standard deviations across 30 different trails) in four different running modes: **Table 2**, Mode A (context free); **Table 3**, mode B (context-sensitive mode); **Table 4**, "semantically supervised" mode,

in which the equivalence classes of the target grammar are made available to the learner ahead of time (training in Mode A); and **Table 5**, bootstrap mode, which starts from a letter-level training corpus in which all spaces between words are omitted (training in Mode A). In the first three experiments, the context-window length was varied while the other parameters were kept fixed ($\eta = 0.6$, $\alpha = 0.01$, corpus size 200). In the bootstrap mode, the algorithm must first segment the sequence of letters into words (applying only the MEX procedure without extracting equivalence classes) and only then use the identified words to extract the grammar. This two-stage process requires a larger corpus to attain a comparable level of performance (up to $10,000$ sentences in this example). Thus, in the last experiment $L$ was kept fixed at 3, $\omega$ was lowered to $0.4$, and the corpus size ranged from $200$ to $10,000$ sentences. Performance was assessed by the F1 measure, defined as $2\cdot$recall$\cdot$precision$/$(recall + precision). The best recall/precision combinations appear in bold and are plotted in Fig. $3A$ in the main paper. It can be seen that both context-free mode and context-sensitive mode reach similar F1 levels; however, while the context-free mode gets higher levels of recall (83% versus 68%) the context-sensitive mode gets higher level of precision (98% versus 80%). When semantic information is available to the learner ahead of time, it gives rise to a significant improvement in the learning performance (F1 = $0.89$ versus $0.81$), which parallels the documented importance of embodiment cues in language acquisition by children. Fig. 8 demonstrates the ability of ADIOS to deal with the kind of syntactic phenomena that can be produced by the TA1 grammar (e.g. "tough movement").

# 3. Learning a Complex CFG

## 3.1. Inferring the ATIS-CFG: supplement to Fig. 3$B$

Table 7 illustrates the recall and precision performance for learning the 4,592-rule ATIS CFG,[†] using different parameter values (L = {3, 4, 5, 6}; 30 or 150 learners; corpus size between 10,000 and 120,000 sentences). Fig. 9 presents a schematic illustration of the coverage of the target language by multiple learners, for various settings of L.

# 4. Further Tests of Grammar Acquisition

## 4.1. Generativity of the learned grammar in natural language: supplement to Fig. 3$C$

Because the target grammar of a natural language is inaccessible, precision must be evaluated by human subjects (referees), while recall can be evaluated by the same method described in the section *Language: Computational Grammar Induction* in the main paper. In the present experiment, the ADIOS algorithm was trained on the ATIS-2 natural language corpus. This corpus contains $13,043$ sentences of natural speech, in the Air Travel Information System (ATIS) domain. The ADIOS algorithm was trained on $12,700$ sentences ($C_{\mathrm{training}}$); the remaining 343 sentences were used to evaluate recall ($C_{\mathrm{target}}$). Two groups of learners (30, 150) were trained ($\eta = 0.6$, $\alpha = 0.01$, $L = 5$) on different, order-permuted, versions of the corpus (several representative acquired patterns appear in Fig. 10 along with their *generalization factors*). After training, each learner generated 100 sentences, which were then placed together into a single corpus (the $C_{\mathrm{learners}}$ test-corpus). Precision of the ADIOS representation (mean $\pm$ SD) was estimated by having eight human subjects judge the acceptability of 20 sentences taken from $C_{\mathrm{learners}}$ and of 20 sentences taken from the original ATIS-2 corpus ($C_{\mathrm{training}}$). The subjects had no indication which sentence belonged to which

[†]Moore, B., & Carroll, J. (2001), *Parser Comparison – Context-Free Grammar (CFG) Data*, http://www.informatics.susx.ac.uk/research/nlp/carroll/cfg-resources.

corpus; the sentences appeared in a random order and each subject judged a different set of sentences. Altogether, 320 sentences were evaluated. The original ATIS-2 corpus was scored at $70 \pm 20\%$ precision while the ADIOS-generated sentences attained $67 \pm 7\%$ precision. Recall was calculated using the $C_{\text{target}}$ corpus. Sets of 30 and 150 learners achieved $32\%$ and $40.5\%$ recall, respectively.

### 4.2. Structured Statistical Language Modeling (SSLM)

The performance of a grammar acquisition algorithm can be assessed by measuring the utility of the acquired representation as a language model, that is, a description of the probabilistic constraints on word order. A model is evaluated on the basis of its ability to predict the next word in a sequence, using sentences taken from a previously unseen text. Models that result in a high average word probability (equivalent to low perplexity[‡]) are considered superior. Standard statistical language models, such as those based on estimated $n$-gram probabilities, are problematic, for two reasons: (1) probability estimates for rare or unseen events are unreliable, and (2) low-$n$ models fail to capture long-range dependencies between words. An experiment described below shows that the grammar learned by ADIOS can be used to construct a simple, yet effective SSLM (2, 3). Because the patterns learned by ADIOS generalize well, and because they capture long-range dependencies, the resulting SSLM achieves an improvement in perplexity on the ATIS-2 corpus over state-of-the-art models, despite requiring much less training data.

Consider an $l$-word sentence $W = w_1 \ldots w_l$, and the parse trees induced over it by the learned grammar; note that a parse tree may be complete ($T_{1,l}$, spanning the entire sentence), or partial ($T_{i,j}$ ; $1 \leq i < j \leq l$). Our goal is to assign a probability $p(w_k|T_{i,k})$ to every sentence prefix $(w_1, \ldots, w_k)$, and to its every possible partial derivation $T_{i,k}$, where $1 \leq i < k$, $1 \leq k \leq l$, and $p(w_k|T_{k-1,k-n}(i))$ is determined by the branching level of $T_{k-1,k-n}$ at $w_k$. In Fig. 11, for example, the branching level at $w_7$ (the next location in the sequence of terminals) is 2, and thus $p(\mathsf{available}|P1855_{1,6}) = p(\mathsf{served}|P1855_{1,6}) = 1/2$. We calculate these probabilities iteratively. At each step, a simple deterministic parser analyzes all the relevant root patterns (including those that span only parts of the prefix), seeking matching parts (see Fig. 11). The number $n$ of matching words defines the *level of history dependence* of the predicted word (this variable dependence may be contrasted with a standard $n$-gram language model, where $n$ is fixed).

The $n$-gram probability $p(w_k|w_{k-n} \ldots w_{k-1})$ can be calculated as follows:

$$p_n(w_k) \doteq p(w_k|w_{k-n} \ldots w_{k-1}) = \sum_{i=1:m} p(w_k|T_{k-1,k-n}(i))p(T_{k-1,k-n}(i)), \qquad \textbf{[7]}$$

where $p(T_{k-1,k-n})$ is the probability of finding the pattern tree $T_{k-1,k-n}$ in the corpus, and $m$ is the number of structures that span $(k-1, k-n)$. For each word in the test set, the parser provides the values of the $n$-gram probability functions $p_n(w_k)$. The final probability is estimated by linear smoothing

$$P(w_k|w_{k-n} \ldots w_{k-1}) = c_1 p_1(w_k) + c_2 p_2(w_k) + \ldots + c_{k-1} p_{k-1}(w_k), \qquad \textbf{[8]}$$

where $\sum_i c_i = 1$; we set $c_i = \frac{i}{\sum_{j=1:k-1} j}$. When multiple learners are used, we average the probabilities they provide for every predicted word at every location, then normalize.

Prior to testing, we weighted the elements of each learner's ADIOS grammar probabilistically using the sentences of the training set, by applying the parser to successive prefixes of each of the sentences in that set.

[‡]Perplexity measures the degree of uncertainly about the next word, averaged over the test set (2). The lower the perplexity, the better the model.

The results were used to assign probabilities to the relevant pattern elements (terminals and equivalence class members). The final probability is the average of the weighted probability and the predicted probability.[§]

The SSLM derived from ADIOS was applied to the ATIS-2 corpus, which contains $13,043$ sentences, of which $11,386$ are unique. Of these, $10,000$ sentences were used to train the SSLM. The estimated probabilities were normalized to sum to $1$ for each word (this experiment involved 30 learners; the probabilities for every predicted word at every location were averaged across learners, then normalized). We also made sure that all the words in the lexicon were assigned nonzero probabilities, to ensure the validity of the ensuing perplexity estimates. Suppose that there are $N$ words in the lexicon, and the model assigns nonzero probabilities to $d$ words ($d < N$). The remaining $N - d$ words are then each assigned a small probability of $p = \epsilon/(N - d)$; all the other probabilities are renormalized to $p_k(1 - \epsilon)$, with $\epsilon = 0.01$.

The resulting perplexity of the ADIOS-based SSLM was $11.5$. This compares favorably with the published figures for regular language grammar induction algorithms (between $30$ and $40$), and for 3-gram models that use sophisticated smoothing ($\approx 14$); (see refs. 3 and 5 – 7).

### 4.3. Languages other than English: supplement to Fig. 3$D$

To visualize the typological relationships of different languages, we consider the pattern spectrum representation, defined as follows. We first list all the significant patterns extracted from the data during the application of the ADIOS algorithm. Each of these consists of elements that belong to one of three classes: patterns (P), equivalence classes (E), and original words or terminals (T) of the tree representation. We next compute the proportions of patterns that are described in terms of these three classes as TT, TE, TP, and so on, as shown in Fig. 12. Comparing the spectra of the six languages, we derive a dendrogram representation of the relative syntactic proximity between them. This is shown in Fig. $3D$. It corresponds well to the expected pattern of typological relationships suggested by classical linguistic analysis (8).

## 5. Language: Psycholinguistics

### 5.1. Learning "nonadjacent dependencies"

Gómez (9) showed that the ability of subjects to learn an artificial language L1 of the form $\{aXd, bXe, cXf\}$, as measured by their ability to distinguish it implicitly from L2=$\{aXe, bXf, cXd\}$, depends on the amount of variation introduced at $X$ (symbols $a$ through $f$ here stand for 3- or 4-letter nonsense words, whereas $X$ denotes a slot in which a subset of 2-24 other nonsense words may appear). Within the ADIOS framework, these nonadjacent dependencies translate into patterns with embedded equivalence classes. We replicated the Gómez study by training ADIOS on 432 strings from L1 (30 learners, $|X| = 2, 6, 12, 24$, $\eta = 0.6$, $\alpha = 0.01$). Training with the context window parameter $L$ set to 3 resulted in performance levels (rejection rate of patterns outside of the learned language) that increased monotonically with $|X|$, in correspondence with the human behavior. Interestingly, when trained with $L = 4$, ADIOS reaches perfect performance in this task.

The two languages used in ref. 9, L1 and L2, are defined in Table 8. *Pel, vot, dak, tood* are all nonsense words that form three-element sequences, in whose middle slot, denoted by $X$, a subset of between 2 and 24 other nonsense words may appear. In the ADIOS terms, $X$ thus stands for an equivalence class with 2-24 elements. We replicated the Gómez study by training ADIOS on 432 strings from L1, using 30 learners and

[§] A more complicated and time-consuming, but probably better, way of producing a Probabilistic CFG (PCFG) out of the ADIOS rules would have been to use the Inside-Outside Algorithm (4).

various sizes of $X$. Performance was evaluated in the same manner as in the Gómez study. The test set consisted of 12 strings: 6 from L1 (which should be accepted) and 6 from L2 (which should be rejected). The results are as follows: when $L$ is set to 3 ($\eta = 0.6$, $\alpha = 0.01$) and $|X|$ is set to $2, 6, 12, 24$ elements, ADIOS accepts all the sentences of L1 while rejecting $14 \pm 27\%$, $50 \pm 17\%$, $86 \pm 14\%$, $82 \pm 17\%$ sentences of L2, respectively. Performance level increases monotonically with $|X|$, in accordance with human data. Training with $L = 4$ yielded $100\%$ acceptance rate for L1 and $100\%$ rejection rate for L2, irrespectively of $|X|$, indicating a perfect ability of the algorithm to capture the nonadjacent dependency rule with the proper choice of parameters.

### 5.2. Grammaticality judgments

A single instance of ADIOS was trained on the CHILDES (10) corpus, using sentences spoken by parents to 3-year-old children. It was then subjected to five grammaticality judgment tests. One of these, the Göteborg multiple-choice ESL (English as Second Language) test, consists of 100 sentences, each containing an open slot; for each sentence, the subject selects one word from a list of three choices, so as to complete the sentence grammatically. In this test, ADIOS scored at $60\%$, which is the average score for 9th grade ESL students. Interestingly, the average score of ADIOS on the entire collection of tests was at the same level.

We have assessed the ability of the ADIOS model to deal with novel inputs[¶] by introducing an *input module* (described below). After training on transcribed speech directed at children [a corpus of 300,000 sentences with $1.3$ million words, taken from the CHILDES collection (10)], the input module was subjected to grammaticality judgment tests, in the form of multiple choice questions. The algorithm[‖] identified 3,400 patterns and 3,200 equivalence classes. The input module was used to process novel sentences by forming their distributed representations in terms of activities of existing patterns [a similar approach had been proposed for novel object and scene representation in vision (11)]. These values, which supported grammaticality judgment, were computed by propagating activation from bottom (the terminals) to top (the patterns). The initial activities $a_j$ of the terminals $e_j$ were calculated given a stimulus $s_1, \ldots, s_k$ as follows:

$$a_j = \max_{l=1..k} \left\{ P(s_l, e_j) \log \frac{P(s_l, e_j)}{P(s_l)P(e_j)} \right\}, \qquad \textbf{[9]}$$

where $P(s_l, e_j)$ is the joint probability of $s_l$ and $e_j$ appearing in the same equivalence class, and $P(s_l)$ and $P(e_j)$ are the probabilities of $s_l$ and $e_j$ appearing in any equivalence class. For an equivalence class, the value propagated upward was the strongest nonzero activation of its members; for a pattern, it was the average weight of the children nodes, on the condition that all the children were activated by adjacent inputs. Activity propagation continued until it reached the top nodes of the pattern lattice. When this algorithm encounters a novel word, all the members of the terminal equivalence class contribute a value of $\epsilon = 0.01$, which is then propagated upward as before. This enables the model to make an educated guess as to the meaning of the unfamiliar word, by considering the patterns that become active. Fig. 13 shows the activation of a pattern (#185) by a sentence that contains a word in a novel context (new), as well as other words never before encountered in any context (Linda, Paul).

We assessed this approach by subjecting a single instance of ADIOS to five different grammaticality judgment tests reported in the literature (12 – 15); see Fig. 14 *Left*. The results of one such test, used in ESL classes, are described below. This test has been administered in Göteborg (Sweden) to $> 10,000$ upper secondary levels students (that is, children who typically had 9 years of school, but only 6-7 years

[¶] Including sentences with novel vocabulary items that are not fully represented by the trained system.

[‖] An earlier version of ADIOS, which did not use the full conditional probability matrix of Eq. **[6]**.

of English). The test consists of $100$ three-choice questions (Table 9), with $65\%$ being the average score for the population mentioned. For each of the three choices in a given question, our algorithm provided a grammaticality score. The choice with the highest score was declared the winner; if two choices received the same top score, the answer was "don't know." The algorithm's performance is plotted in Fig. 14 *Right* against the size of the CHILDES training set. Over the course of training, the proportion of questions that received a definite answer grew (red bars), while the proportion of correct answers remained around $60\%$ (blue curve); compare this to the $45\%$ precision with $20\%$ recall achieved by a straightforward bi-gram benchmark.[**]

# 6. Bioinformatics

## 6.1. Classification of enzymes classes

We evaluated the ability of root patterns found by ADIOS to support functional classification of proteins (enzymes). The function of an enzyme is specified by an Enzyme Commission (EC) name. The name corresponds to an EC number, which is of the form: n1:n2:n3:n4. In this experiment, we concentrated on the oxidoreductases superfamily (EC 1.x.x.x). Protein sequences and their EC number annotations were extracted from the SwissProt database Release 40.0; sequences with double annotations were removed. First, ADIOS was loaded with all the 6751 proteins of the oxidoreductases superfamily. Each path in the initial graph thus corresponded to a sequence of amino acids (20 symbols).

The training stage consisted of the two-stage action described in section 2.2. In the first stage ($\eta = 0.9$, $\alpha = 0.01$), the algorithm identified $10,200$ motifs (words). In the second stage ($\eta = 1.0$, $\alpha = 0.01$) after removing those letters that were not associated with one of the identified motifs, it extracted additional 938 patterns. Classification was tested on level 2 (EC 1.x, 16 classes) and on level 3 (EC 1.x.x, 54 classes). Proteins were represented as vectors of ADIOS root patterns. A linear SVM classifier (SVM-LIGHT package, available online at http://svmlight.joachims.org/) was trained on each class separately, taking the proteins of the class as positive examples, and the rest as negative examples. Seventy-five percent of the examples were used for training and the remainder for testing. Performance was measured as $Q = (TP + TN)/(TP + TN+FP+FN)$, where $TP$, $TN$, $FP$, and $FN$ are, respectively, the number of true positive, true negative, false positive, and false negative outcomes. Table 10 presents the performance of the ADIOS algorithm on level 2 alongside the performance of the SVM-PROT system (16); Table 11 presents the performance on level 3. The ADIOS performance matched the performance of the SVM-PROT system (Fig. 15$A$), even though the latter uses a representation composed of features such as hydrophobicity, normalized van der Waals volume, polarity, polarizability, charge, surface tension, secondary structure and solvent accessibility, whereas we use solely the structure found by our algorithm in the amino acid sequence data. The average recall/precision on level 2 was $71 \pm 13\%$ and $90 \pm 9\%$, respectively, whereas recall/precision on level 3 was $70 \pm 26\%$ and $93 \pm 23\%$, indicating that the ADIOS representation can accurately discriminate the enzyme's low-level functionality.

## 6.2. A compression ratio analysis

Our algorithm provides a useful tool for identifying open reading frames (ORF) and coding regions in DNA sequences, based on comparing the description length of the ADIOS representation before and after learning.

[**] Chance performance in this test is $33\%$. We note that the corpus used here was too small to train an $n$-gram model for $n > 2$; thus, our algorithm effectively overcomes the problem of sparse data by putting the available data to a better use.

The description length of the ADIOS representation consists of two parts: the graph (vertices and paths) and the identified patterns. The *compression ratio* of the description length can be quantified by evaluating the decrease in the physical memory it occupies (in bits).

Following its iterative application, ADIOS compresses the initial graph to a final graph plus a forest of distilled root-patterns. Although the latter can generate a much larger corpus than the original one, the description length of the ADIOS representation is diminished. The recall level of ADIOS increases with compression. Applying ADIOS to the coding region of the *Caenorhabditis elegans* genome (Fig. 15$B$), we conclude that the syntax of ORF0 (the correct open reading frame) is the one to be preferred. Moreover, we are able to distinguish between coding and non-coding regions (Fig. 15$B$ vs. 15$C$), because for the latter different ORFs lead to similar compression levels.

We have also calculated the compression at several points along the curves of the ATIS-CFG recall/precision graph (Fig. 3$B$). Fig. 16 shows the correlation between the recall/precision levels (ordinate) and the compression rate (abscissa). It can be seen that ADIOS recall level strongly depends on (increases with) the compression level, but the precision level only weakly depends on the latter. The compression ratio characteristic is particularly useful when comparing the performance of ADIOS on different data for which the target "grammars" are not available. The ORF problem is a typical example of such an analysis.

## 7. Computational Complexity

We conducted several experiments based on the TA1 grammar to estimate the computational complexity of ADIOS. We found four variables that have major effects: the total number of words in a given corpus, the average sentence length, the size of the initial lexicon and the value of the context window parameter L. For each of these, we conducted an experiment that exclusively manipulated the variable in question, while measuring the time until convergence. The results, plotted in Fig. 17, reveal the following dependencies: the training time grows linearly with the size of the corpus and logarithmically with the average sentence length. It shows inverse power dependence both on respect the lexicon size and on the value of L. Overall, the computational complexity of ADIOS according to this empirical estimate is $O\left(nlog\left(l\right)/\left(L^{\lambda}N^{\gamma}\right)\right)$, where $n$ is the total number of words in the corpus, $l$ is the average sentence length, $L$ is the value of context window parameter, and $N$ is the lexicon size. The conclusion from this experiment is that ADIOS is easily scalable to larger corpora; this finding is consistent with the actual tests described in the main text.

## 8. Conclusions

The massive, largely unsupervised, effortless, and fast feat of learning that is the acquisition of language by children has long been a daunting challenge for cognitive scientists (17, 18) and for natural language engineers (19 – 21). Because a completely bias-free unsupervised learning is impossible (17, 22, 23), the real issue in language acquisition is to determine the constraints that a model of "grammar induction" should impose — and to characterize those constraints that infants acquiring language do in fact impose — on the learning procedure. In our approach, the constraints are defined algorithmically, in the form of a method for detecting, in sequential symbolic data, of units (patterns and equivalence classes) that are hierarchically structured and are supported by context-sensitive statistical evidence.

In linguistics, our method should be of interest to researchers of various theoretical persuasions who construe grammars as containing — in addition to general and lexicalized (24, 25) rules — "inventories" of units of varying kinds and sizes (26, 27) such as: idioms and semiproductive forms (28, 29), prefabricated

expressions (30, 31), "syntactic nuts" (32), frequent collocations (33), multiword expressions (34, 35), and constructions (36 – 39). In addition, the growing collection of patterns revealed by our algorithm in various corpora should complement both syntax-related resources such as the Penn Treebank (40) and semantics-oriented resources such as the WordNet (41), the PhraseNet (42), and the Berkeley FrameNet (43, 44).

## References


1. Adriaans, P & Vervoort, M. (2002) in *Grammatical Inference: Algorithms and Applications: 6th International Colloquium: ICGI 2002*, Lecture Notes in Computer Science, eds. Adriaans, P, Fernau, H, & van Zaanen, M. (Springer-Verlag, Heidelberg) Vol. 2484, pp. 293–295.

2. Goodman, J. T. (2001) A bit of progress in language modeling: Extended version, (Microsoft Research), Technical Report MSR-TR-2001-72.

3. Chelba, C. (2001) in *Proc. of the Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*. (Salt Lake City, UT) Vol. 1, pp. 544a–544d.

4. Lari, K & Young, S. (1990) *Computer Speech and Language* **4**, 35–56.

5. McCandless, M & Glass, J. (1993) in *Proc. EuroSpeech'93*. pp. 981–984.

6. Roark, B. (2001) *Computational Linguistics* **27**, 249–276.

7. Kermorvant, C, de la Higuera, C, & Dupont, P. (2004) *Journal électronique d'intelligence artificielle* **6**.

8. Grimes, P. (2001) *Data from Ethnologue: Languages of the World (14th Edition)*. (SIL International).

9. Gómez, R. L. (2002) *Psychological Science* **13**, 431–436.

10. MacWhinney, B & Snow, C. (1985) *Journal of Computational Lingustics* **12**, 271–296.

11. Edelman, S. (2002) *Trends in Cognitive Sciences* **6**, 125–131.

12. Linebarger, M. C, Schwartz, M, & Saffran, E. (1983) *Cognition* **13**, 361–392.

13. Lawrence, S, Giles, C. L, & Fong, S. (2000) *IEEE Transactions on Knowledge and Data Engineering* **12**, 126–140.

14. Allen, J & Seidenberg, M. S. (1999) in *Emergence of Language*, ed. MacWhinney, B. (Lawrence Earlbaum Associates, Hillsdale, NJ).

15. Martin, R. C & Miller, M. D. (2002) in *Handbook of Adult Language Disorders: Integrating Cognitive Neuropsychology, Neurology, and Rehabilitation*, ed. Hillis, A. (Psychology Press, New York).

16. Cai, C. Z, Han, L. Y, Ji, Z. L, Chen, X, & Chen, Y. Z. (2003) *Nucleic Acids Research* **31**, 3692–3697.

17. Chomsky, N. (1986) *Knowledge of language: its nature, origin, and use*. (Praeger, New York).

18. Elman, J. L, Bates, E. A, Johnson, M. H, Karmiloff-Smith, A, Parisi, D, & Plunkett, K. (1996) *Rethinking innateness: A connectionist perspective on development*. (MIT Press, Cambridge, MA).

19. Bod, R. (1998) *Beyond grammar: an experience-based theory of language*. (CSLI Publications, Stanford, US).

20. Clark, A. (2001) Ph.D. thesis (COGS, University of Sussex).

21. Roberts, A & Atwell, E. (2002) Unsupervised grammar inference systems for natural language, (School of Computing, University of Leeds), Technical Report 2002.20.

22. Nowak, M. A, Komarova, N. L, & Niyogi, P. (2001) *Science* **291**, 114–119.

23. Baum, E. B. (2004) *What is thought?* (MIT Press, Cambridge, MA).

24. Daumé, H, Knight, K, Langkilde-Geary, I, Marcu, D, & Yamada, K. (2002) *The importance of lexicalized syntax models for natural language generation tasks*. (Harriman, NY), pp. 9–16.

25. Geman, S & Johnson, M. (2003) in *Mathematical foundations of speech and language processing*, IMA Volumes in Mathematics and its Applications, eds. Johnson, M, Khudanpur, S, Ostendorf, M, & Rosenfeld, R. (Springer-Verlag, New York) Vol. 138, pp. 1–26.

26. Langacker, R. W. (1987) *Foundations of cognitive grammar*. (Stanford University Press, Stanford, CA) Vol. I: theoretical prerequisites.

27. Daelemans, W. (1998) in *English as a human language*, eds. van der Auwera, J, Durieux, F, & Lejeune, L. (LINCOM Europa, Munchen), pp. 73–82.

28. Jackendoff, R. (1997) *The Architecture of the Language Faculty*. (MIT Press, Cambridge, MA).

29. Erman, B & Warren, B. (2000) *Text* **20**, 29–62.

30. Makkai, A. (1995) in *Syntactic iconicity and linguistic freezes*, ed. Landsberg, M. E. (Mouton de Gruyter, Berlin), pp. 91–116.

31. Wray, A. (2000) in *The evolutionary emergence of language*, eds. Knight, C, Studdert-Kennedy, M, & Hurford, J. R. (Cambridge University Press, Cambridge), pp. 285–302.

32. Culicover, P. W. (1999) *Syntactic nuts: hard cases, syntactic theory, and language acquisition*. (Oxford University Press, Oxford).

33. Bybee, J & Hopper, P. (2001) in *Frequency and the emergence of linguistic structure*, eds. Bybee, J & Hopper, P. (John Benjamins, Amsterdam), pp. 1–24.

34. Sag, I. A, Baldwin, T, Bond, F, Copestake, A, & Flickinger, D. (2002) in *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002)*. (Mexico City, Mexico), pp. 1–15.

35. Baldwin, T, Bannard, C, Tanaka, T, & Widdows, D. (2003) in *Proceedings of the ACL-2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. (Sapporo, Japan), pp. 89–96.

36. Kay, P & Fillmore, C. J. (1999) *Language* **75**, 1–33.

37. Croft, W. (2001) *Radical Construction Grammar: syntactic theory in typological perspective*. (Oxford University Press, Oxford).

38. Goldberg, A. E. (2003) *Trends in Cognitive Sciences* **7**, 219–224.

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

40. Marcus, M. P, Santorini, B, & Marcinkiewicz, M. A. (1994) *Computational Linguistics* **19**, 313–330.

41. Miller, G. A & Fellbaum, C. (1991) *Cognition* **41**, 197–229.

42. Li, X, Roth, D, & Tu, Y. (2003) in *Proceedings of CoNLL-2003*, eds. Daelemans, W & Osborne, M. (Edmonton, Canada), pp. 87–94.

43. Baker, C. F, Fillmore, C. J, & Lowe, J. B. (1998) in *Proceedings of the COLING-ACL*. (Montreal, Canada) Vol. 1, pp. 86–90.

44. Baker, C. F, Fillmore, C. J, & Cronin, B. (2003) *International Journal of Lexicography* **16**, 281–296.