Supporting online material

Zach Solan, David Horn, Eytan Ruppin, Shimon Edelman

September 21, 2004

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 non-simple 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. Figure S1 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* $\mathbf{S} = (e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_k) = (e_1; e_k)$.[1] 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 sub-path $e_i \rightarrow e_{i+1} \rightarrow e_{i+2}... \rightarrow e_{j-1}$:

$$P_R(e_i; e_j) = p(e_j|e_i e_{i+1} e_{i+2}...e_{j-1}) = \frac{l(e_i; e_j)}{l(e_i; e_{j-1})} \tag{1}$$

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

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

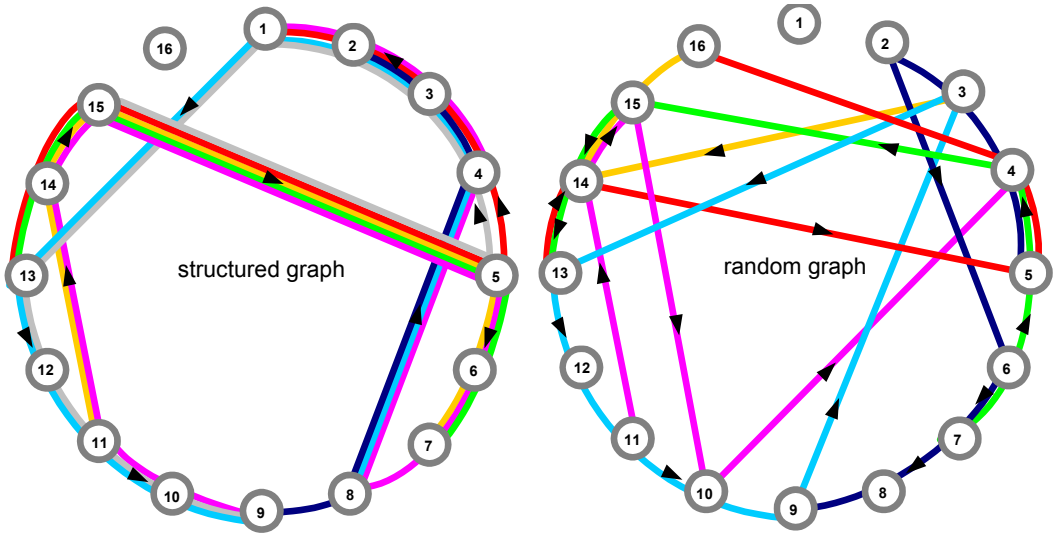

Figure S1: Comparison between a structured graph, of the type expected to appear in our problem (left), and one of random connectivity (right).

function P_L:

$$P_L(e_j; e_i) = p(e_i|e_{i+1}e_{i+2}...e_{j-1}e_j) = \frac{l(e_j; e_i)}{l(e_j; e_{i+1})} \tag{2}$$

and note that

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

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 Figure S2, 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 $D_L(e_j; e_i) = P_L(e_j; e_i)/P_L(e_{j+1}; e_i) < \eta$ 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:

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

We calculate both P_L and P_R from all the possible starting points (such as $e1$ and $e4$ in the example of Figure S2), traversing each path leftward and rightward, correspondingly. This defines a matrix of the form

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

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:

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

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}).

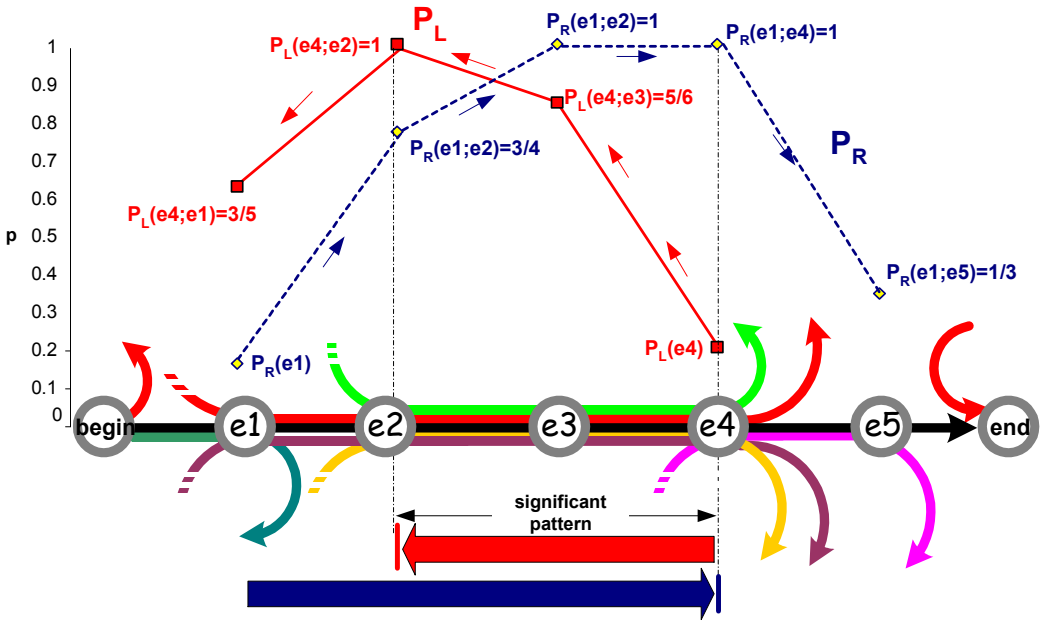

Figure S2: The definition of a bundle that serves as a candidate pattern, whose beginning and end are signaled by the maxima of P_L and P_R. It becomes a candidate because of the large drops in these probabilities after the maxima, signifying the divergence of paths at these points.

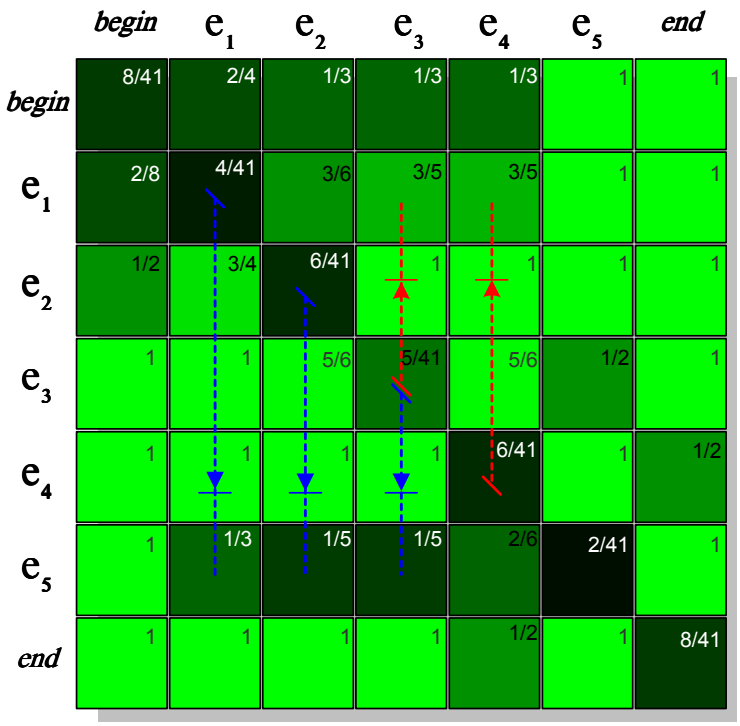

Figure S3: An instance of a 7×7 M matrix based on the black search path in Figure S2. The blue and red arrows represent all the significant segments $D_R(e_a; e_b)$ and $D_L(e_d; e_c)$ ($\alpha < 0.01$), respectively. The values of the matrix elements appear in the upper right corners of the cells. The most significant pair of segments ($B(e_a; e_b)$, $B(e_d; e_c)$) for which $a < d < b < c$ is marked the *leading pattern* (in this example the leading pattern is $e_2 \rightarrow e_3 \rightarrow e_4$).

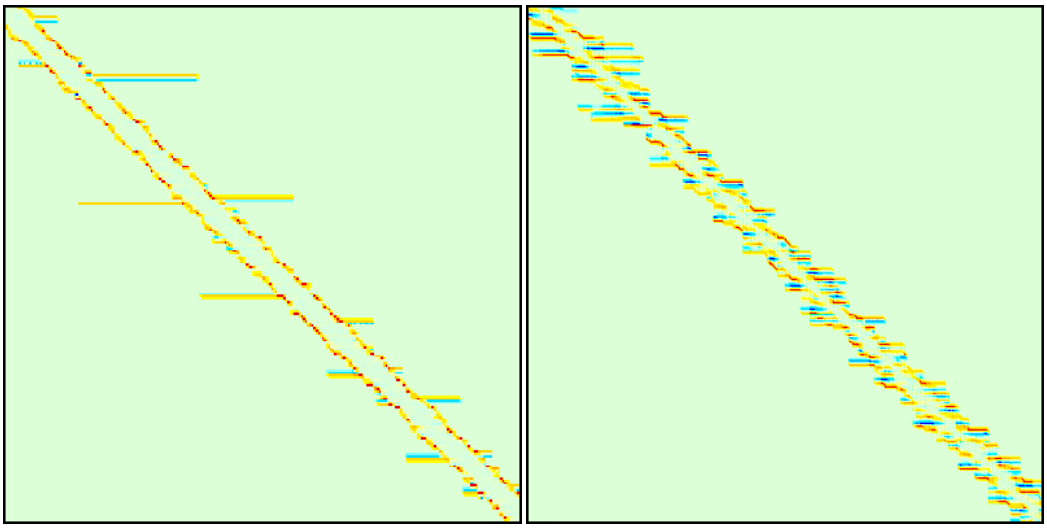

Figure S4: Two instances of the M matrix computed from two different corpora: the SwissProt database sequence of amino acids (*left*), and the text of *Alice in the Wonderland* considered as a sequence of letters (*right*). Significant changes in the P_L and P_R values have been colored on a yellow/red scale and a cobalt/blue scale (increase and decrease, respectively); green is the neutral color. The protein O17433 is the search path used to construct the matrix on the left; the first paragraph of *Alice* is the search path used to create the matrix on the right. For visualization purposes, only the first 300 elements of the matrices are shown.

2 Learning a simple Context-Free Grammar

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

We replicated one of the experiments of (*1*) ("A 2000 Sentences Sample", p.8). The aim of the original experiment was to reconstruct a specific context-free grammar (29 terminals and 7 rules) from a corpus of 2000 sentences using the EMILE 4.1 algorithm. The results of applying the ADIOS algorithm to a 2000-sentence corpus randomly generated from the given context-free grammar are shown in Table S1. 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 (*1*), induced 3000-4000 rules (the recall/precision performance of the EMILE algorithm was not stated). Table S1 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 Figure 3A

Tables S2 to S5 show the performance of an ADIOS model trained on extremely small corpora (200 sentences) generated by the TA1 artificial grammar (listed in Table S6). The tables present the recall-precision values (with their standard deviations across 30 different trails) in four different running modes: **Table S2**, Mode A (context free); **Table S3**, mode B (context-sensitive mode); **Table S4**, "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); **Table S5**, 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

Table S1: A comparison between the target grammar of Adriaans and Vervoort (*left*) and the grammar induced by a single ADIOS instance (*right*). Root patterns appear in bold.

target grammar	**inferred grammar**
[NP_a] ⇒ John | Mary | the man | the child	E35 ⇒ child | man P34 ⇒ the E35 E37 ⇒ John | Mary | P34
[P] ⇒ with | near | in | from	E54 ⇒ with | near | in | from
[V_i] ⇒ appears | is | seems | looks	E39 ⇒ appears | is | seems | looks
[V_s] ⇒ thinks | hopes | tells | says	E51 ⇒ thinks | hopes | tells | says
[V_t] ⇒ knows | likes | misses | sees	E46 ⇒ knows | likes | misses | sees
[ADV] ⇒ large | small | ugly | beautiful	E49 ⇒ large | small | ugly | beautiful
[NP_p] ⇒ the car | the city | the house | the shop	E43 ⇒ house | shop
[S] ⇒ [NP] [V_i] [ADV] | [NP_a] [VP_a] | [NP_a] [V_s] that [S] [NP] ⇒ [NP_a] | [NP_p] [VP_a] ⇒ [V_t] [NP] | [V_t] [NP] [P] [NP_p]	E69 ⇒ P47 | P59 | P62 | P66 | P67 P40 ⇒ the city P41 ⇒ the car **P36** ⇒ John likes E37 P42 ⇒ the E43 E39 P44 ⇒ the house P45 ⇒ E37 E46 **P47** ⇒ P45 E37 **P48** ⇒ E37 E39 E49 P50 ⇒ E37 E51 that **P52** ⇒ P47 in the shop P53 ⇒ E54 P44 **P55** ⇒ P47 near P40 P56 ⇒ E54 P41 P57 ⇒ E54 P40 **P58** ⇒ P50 P50 P45 the shop **P59** ⇒ P45 the shop **P60** ⇒ P41 E39 E49 **P61** ⇒ P42 E49 **P62** ⇒ P45 P40 **P63** ⇒ P50 P50 P48 P64 ⇒ E54 the shop **P65** ⇒ P50 P62 P64 **P66** ⇒ P45 P44 **P67** ⇒ P45 P41 **P68** ⇒ E69 P53 **P70** ⇒ P38 E49 **P71** ⇒ E69 P57

Table S2: Mode A (Context-Free).

corpus size	L	recall	precision	F1
200	9	0.3 ± 0.2	0.9 ± 0.1	0.45
200	8	0.4 ± 0.2	0.93 ± 0.09	0.59
200	7	0.6 ± 0.1	0.9 ± 0.1	0.71
200	6	0.7 ± 0.1	0.9 ± 0.2	0.78
200	5	0.78 ± 0.08	0.8 ± 0.2	0.80
200	**4**	$\mathbf{0.83 \pm 0.06}$	$\mathbf{0.8 \pm 0.2}$	**0.81**
200	3	0.84 ± 0.06	0.6 ± 0.2	0.71

Table S3: Mode B (Context-Sensitive).

corpus size	L	recall	precision	F1
200	9	0.5 ± 0.2	0.8 ± 0.1	0.65
200	8	0.6 ± 0.1	0.78 ± 0.09	0.66
200	7	0.61 ± 0.07	0.9 ± 0.2	0.72
200	6	0.6 ± 0.1	0.8 ± 0.2	0.68
200	5	0.61 ± 0.09	0.8 ± 0.1	0.69
200	4	0.69 ± 0.05	0.9 ± 0.1	0.79
200	**3**	$\mathbf{0.68 \pm 0.06}$	$\mathbf{0.98 \pm 0.04}$	**0.80**

Table S4: Mode A, “semantically supervised”.

corpus size	L	recall	precision	F1
200	8	0.86 ± 0.06	0.7 ± 0.2	0.80
200	7	0.89 ± 0.04	0.8 ± 0.2	0.84
200	6	0.90 ± 0.04	0.8 ± 0.2	0.85
200	5	0.90 ± 0.03	0.8 ± 0.2	0.83
200	4	0.92 ± 0.03	0.8 ± 0.2	0.83
200	**3**	$\mathbf{0.92 \pm 0.03}$	$\mathbf{0.9 \pm 0.2}$	**0.89**

Table S5: Mode A, “no spaces”.

corpus size	L	recall	precision	F1
200	3	0.01 ± 0.01	0.91 ± 0.09	0.01
500	3	0.07 ± 0.04	0.89 ± 0.08	0.12
1000	3	0.13 ± 0.06	0.8 ± 0.1	0.23
2500	3	0.30 ± 0.07	0.79 ± 0.09	0.43
5000	3	0.39 ± 0.08	0.85 ± 0.1	0.53
10000	3	0.5 ± 0.1	0.86 ± 0.09	0.65

Table S6: The TA1 grammar, consisting of 50 terminals and 28 rules.

σ	$\Rightarrow$ s1 | s2 | s3 | s4
s1	$\Rightarrow$ prec np2 vp ptag
s2	$\Rightarrow$ frec np2 vp ftag
s3	$\Rightarrow$ frec iv6 iv55
s4	$\Rightarrow$ *that* np2 iv5 iv6 iv4 np2
np	$\Rightarrow$ art noun | propn
np2	$\Rightarrow$ *the* noun | propn
propn	$\Rightarrow$ p vp2 | p
pp	$\Rightarrow$ p *and* p vp6 | p p *and* p vp6
vp	$\Rightarrow$ iv *and* com
vp2	$\Rightarrow$ *who* tv np
com	$\Rightarrow$ np iv2
rec	$\Rightarrow$ p vp5 *that* rec | p vp5 *that*
frec	$\Rightarrow$ pf vp5 *that* rec
ftag	$\Rightarrow$ *, doesn't she ?*
prec	$\Rightarrow$ pp *that* rec
ptag	$\Rightarrow$ *, don't they ?*
iv5	$\Rightarrow$ *is* iv5-ex
iv55	$\Rightarrow$ *is* iv55-ex
iv6	$\Rightarrow$ *to* iv6-ex
art	$\Rightarrow$ *the* | *a*
noun	$\Rightarrow$ *cat* | *dog* | *horse* | *cow* | *rabbit* | *bird*
p	$\Rightarrow$ *Joe* | *Beth* | *Jim* | *Cindy* | *Pam* | *George*
pf	$\Rightarrow$ *Beth* | *Cindy* | *Pam*
vp5	$\Rightarrow$ *believes* | *thinks*
vp6	$\Rightarrow$ *believe* | *think*
iv	$\Rightarrow$ *meows* | *barks*
iv2	$\Rightarrow$ *laughs* | *jumps* | *flies*
iv5-ex	$\Rightarrow$ *easy* | *tough* | *eager*
iv55-ex	$\Rightarrow$ *easy* | *tough*
iv6-ex	$\Rightarrow$ *please* | *read*
iv4	$\Rightarrow$ *annoys* | *worries* | *disturbs* | *bothers*
tv	$\Rightarrow$ *scolds* | *loves* | *adores* | *worships*

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, ω 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 $2{\cdot}\text{recall}{\cdot}\text{precision}/(\text{recall} + \text{precision})$. The best recall/precision combinations appear in bold and are plotted in Figure 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. Figure S5 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 Context-Free Grammar

3.1 Inferring the ATIS-CFG: supplement to Figure 3B

Table S7 illustrates the recall and precision performance for learning the 4592-rule ATIS Context Free Grammar (*2*), using different parameter values (L = $\{3, 4, 5, 6\}$; 30 or 150 learners; corpus size between 10,000 and 120,000 sentences). Figure S6 presents a schematic illustration of the coverage of the target language by multiple learners, for various settings of L.

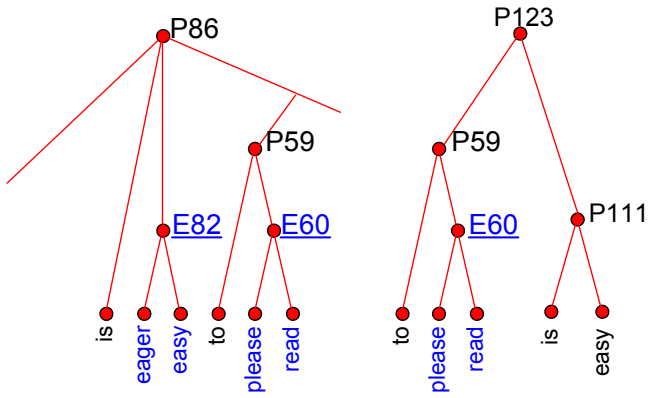

Figure S5: An illustration of the ability of ADIOS to deal with certain structure-dependent syntactic phenomena. In this example, when trained on sentences exemplifying the so-called "tough movement", ADIOS forms patterns that represent the correct phrases (. . . is easy to read, is easy to please, is eager to read, is eager to please, to read is easy and to please is easy), but does not over-generalize to the incorrect ones (*to read is eager or *to please is eager).

4 Generativity of the learned grammar in natural language: supplement to Figure 3C

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-N 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 ($C_{training}$); the remaining 343 sentences were used to evaluate recall (C_{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 Figure S7 along with their *generalization factors*). After training, each learner generated 100 sentences, which were then placed together into a single corpus (the $C_{learners}$ test-corpus). Precision of the ADIOS representation (mean $\pm$ std

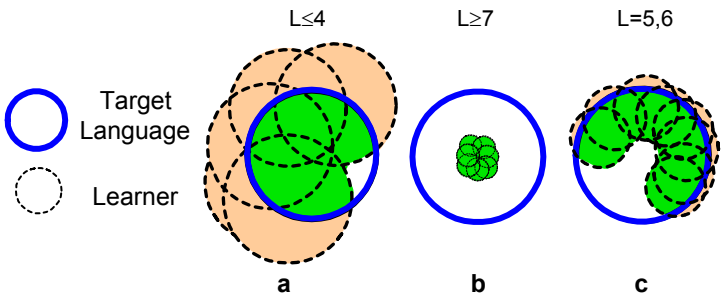

Figure S6: High values of L bring the learners into a low-productivity region where precision is high, but the coverage of the target language is low. For low values of L, the learners tend to over-generalize and thus acquire an inaccurate language that, however, does cover most of the target. The proper balance is achieved by setting L to an intermediate value, so that the learners cover a large portion of the target language, yet remain within the language boundaries.

Table S7: ATIS-CFG recall and precision.

corpus size	No. of learners	L	η	recall	precision	F1
10000	30	3	0.6	0.380	0.623	0.472
10000	30	4	0.6	0.308	0.657	0.420
10000	30	5	0.6	0.180	0.920	0.301
40000	30	3	0.6	0.643	0.568	0.603
40000	30	4	0.6	0.660	0.596	0.627
40000	30	5	0.6	0.456	0.780	0.576
120000	30	3	0.6	0.910	0.538	0.676
120000	30	4	0.6	0.750	0.580	0.654
120000	30	5	0.6	0.747	0.640	0.689
120000	30	6	0.6	0.465	0.818	0.593
120000	150	3	0.6	1.000	0.538	0.700
120000	150	4	0.6	1.000	0.580	0.734
120000	150	5	0.6	1.000	0.640	0.780
120000	150	6	0.6	0.600	0.820	0.693
120000	150	7	0.6	0.230	0.970	0.372

dev) was estimated by having eight human subjects judge the acceptability of 20 sentences taken from $C_{learners}$ and of 20 sentences taken from the original ATIS-N corpus ($C_{training}$). The subjects had no indication which sentence belonged to which 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-N corpus was scored at $70 \pm 20\%$ precision while the ADIOS-generated sentences attained $67 \pm 7\%$ precision. Recall was calculated using the C_{target} corpus. Sets of 30 and 150 learners achieved 32% and 40.5% recall respectively.

4.1 Languages other than English: supplement to Figure 3D

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 Figure S8. Comparing the spectra of the six languages, we derive a dendrogram representation of the relative syntactic proximity between them. This is shown in Figure 3D in the main paper. It corresponds well to the expected pattern of typological relationships suggested by classical linguistic analysis (*3*).

5 Language: psycholinguistics

5.1 Learning "nonadjacent dependencies"

The two languages used in (*4*), L1 and L2, are defined in Table S8. *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

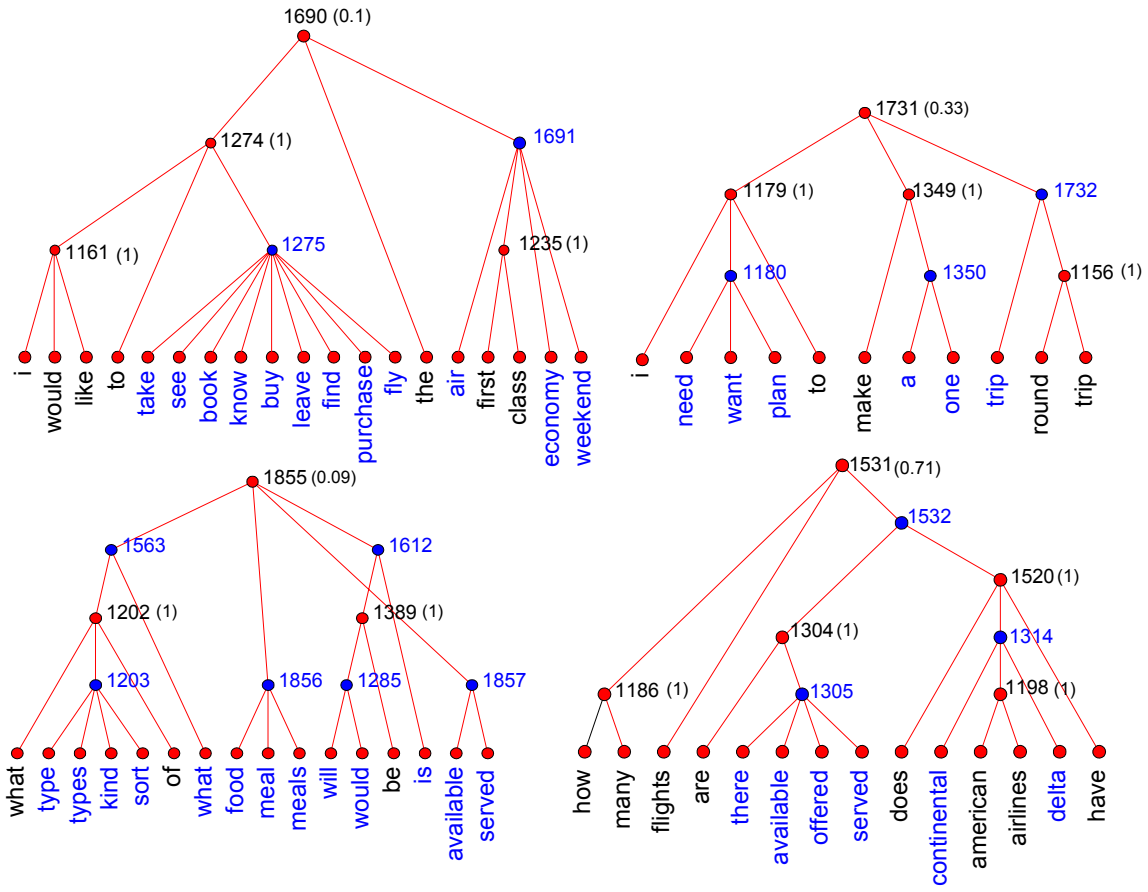

Figure S7: Four simple patterns extracted from the ATIS natural language corpus. Some of the sentences that can be described/generated by patterns #1690, #1731, #1855 and #1531 are: I would like to book the first class; I plan to make a round trip; what kind of food would be served ; how many flights does continental have . None of these sentences appear in the training data, illustrating the ability of ADIOS to generalize. The numbers in parentheses denote the generalization factors of the patterns and their components (e.g., pattern #1690 generates 90% new strings, while pattern #1731 generates 66% new strings).

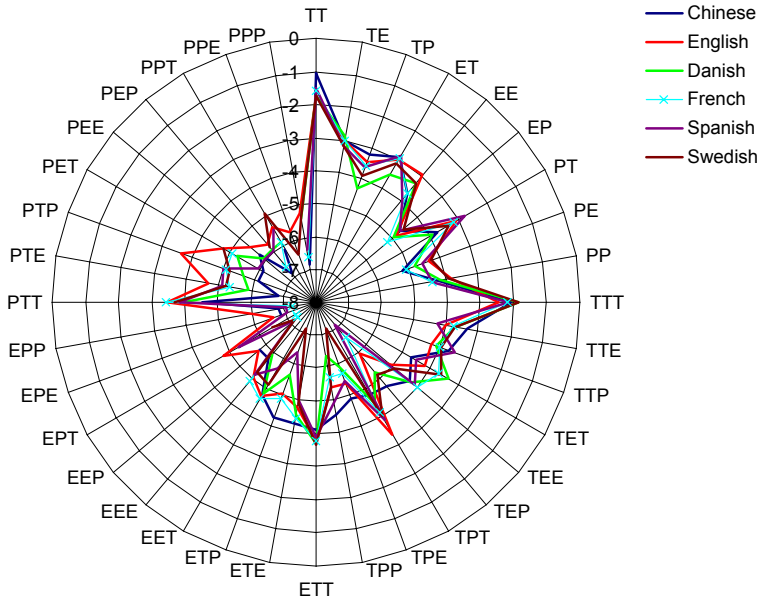

Figure S8: A comparison (on a logarithmic scale) of pattern spectra obtained from six languages (Chinese, English, Danish, French, Spanish and Swedish). The extraction of patterns was based on the same corpus, the Bible (66 books containing 33K sentences), in its six versions, available online at http://www.umiacs.umd.edu/ resnik/parallel/bible.html. It can be seen that natural languages have a relatively large percentage of patterns that fall into TT and TTT categories (known as collocations).

L1	**L2**
$P1 \Rightarrow pel - \mathbf{X} - rud$	$P1 \Rightarrow pel - \mathbf{X} - jic$
$P2 \Rightarrow vot - \mathbf{X} - jic$	$P1 \Rightarrow vot - \mathbf{X} - tood$
$P3 \Rightarrow dak - \mathbf{X} - tood$	$P1 \Rightarrow dak - \mathbf{X} - rud$

X(2)={wadim, kicey}
X(6)={wadim, kicey, puser, fengle, coomo, loga}
X(12)={wadim, kicey, puser, fengle, coomo, loga, gople, taspu, hiftam, benez, vamey, skiger}
X(24)={wadim, kicey, puser, fengle, coomo, loga, gople, taspu, hiftam, benez, vamey, skiger, benez, gensim, feenam, laeljeen, chila, roosa, plizet, balip, malsig, suleb, nilbo, wiffle}

Table S8: Definition of the L1 and L2 languages.

on 432 strings from L1, using 30 learners and 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 non-adjacent dependency rule with the proper choice of parameters.

5.2 Grammaticality judgments

We have assessed the ability of the ADIOS model to deal with novel inputs[2] 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 (*5*)), the input module was subjected to grammaticality judgment tests, in the form of multiple choice questions. The algorithm[3] identified 3400 patterns and 3200 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 (*6*)). 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:

$$a_j = \max_{l=1..k} \left\{ P(s_l, e_j) \log \frac{P(s_l, e_j)}{P(s_l)P(e_j)} \right\} \tag{6}$$

where $P(s_l, e_j)$ is the joint probability of s_l and e_j appearing in the same equivalence class,

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

[3]An earlier version of ADIOS, which did not use the full conditional probability matrix of eq. 1.

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 non-zero 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. Figure S9 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 (*7–10*); see Figure S10 (left). The results of one such test, used in English as Second Language (ESL) classes, are described below. This test has been administered in Göteborg (Sweden) to more than $10,000$ upper secondary levels students (that is, children who typically had 9 years of school, but only 6-7 years of English). The test consists of 100 three-choice questions (Table S9), 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 Figure S10 (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.[4]

[4]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.

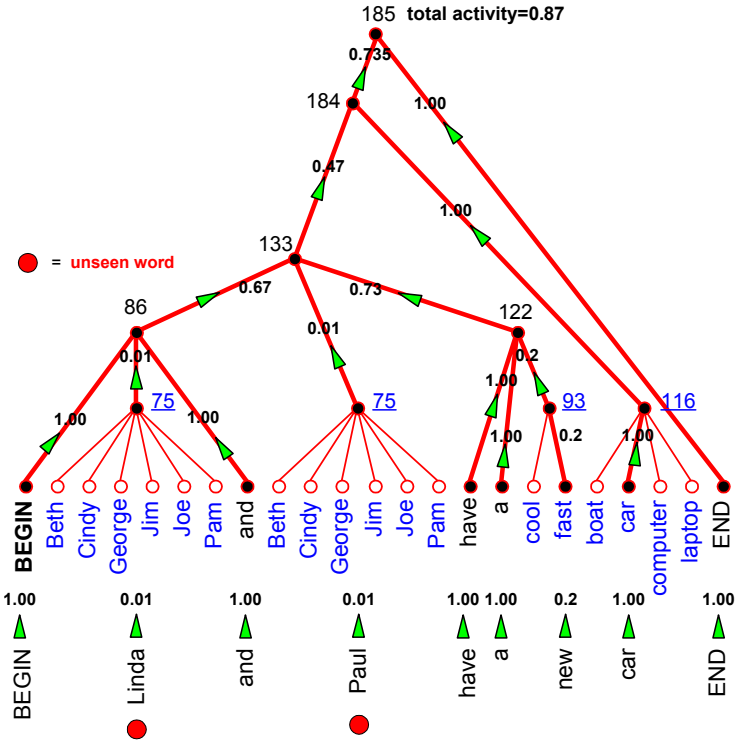

Figure S9: An active pattern responding to the partially novel input Linda and Paul have a new car. Leaf activation, which is proportional to the mutual information between input words and various members of the equivalence classes, is propagated upward by taking the average at each junction.

sentence	*choice 1*	*choice 2*	*choice 3*
The pilot _ look worried.	isn't	doesn't	don't
She asked me _ at once.	come	to come	coming
The tickets have been paid for, so you _ not worry.	may	dare	need
We've gone slightly _ course.	of	off	from

Table S9: Sample questions from a multiple-choice test used in ESL instruction in Göteborg, Sweden. A score $< 50\%$ in this 100-question test (available online) is considered pre-intermediate, $50 - 70\%$ intermediate, and a score $> 70\%$ advanced.

		ADIOS		bi-gram	
benchmark	*#item*	*correct*	*answered*	*correct*	*answered*
Linebarger, Schwartz and Saffran, 1983	25	65%	65%	42%	92%
Lawrence, Giles and Fong, 2000	70	59%	73%	38%	63%
Allen and Seidenberg, 1999	10	83%	60%	40%	50%
Martin and Miller, 2002	10	75%	80%	67%	60%
Goteborg/ESL	100	58%	57%	45%	20%
	215	61%	64%	43%	46%

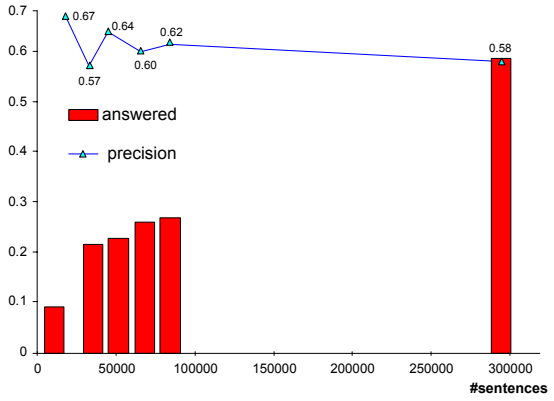

Figure S10: *Left:* the results of several grammaticality tests reported in the literature. *Right:* a summary of the performance of ADIOS in the Göteborg ESL test, plotted against the number of sentences (paths) scanned during training (red bars: recall; blue rectangles: precision).

6 Bioinformatics

6.1 Classification of enzymes classes: supplement to Figure 4A

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. 75% 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 S10 presents the performance of the ADIOS algorithm on level 2 alongside the performance of the SVM-PRot system (*11*); Table S11 presents the performance on level 3. The ADIOS performance matched the performance of the SVM-PRot system, 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, while 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, while 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: supplement to Figure 4B,C

Our algorithm also 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. The description length of the ADIOS representation consists

Class	TP	FN	FP	TN	ADIOS Q	recall	precision	SVM-Prot Q
1.1	333	89	64	1201	0.91	0.79	0.84	0.92
1.2	110	49	26	1502	0.96	0.69	0.81	0.99
1.3	62	36	14	968	0.95	0.63	0.82	0.98
1.4	33	23	9	556	0.95	0.59	0.79	0.99
1.5	20	19	8	384	0.94	0.51	0.71	0.97
1.6	198	23	25	1441	0.97	0.90	0.89	0.99
1.7	23	13	2	365	0.96	0.64	0.92	0.99
1.8	51	21	3	717	0.97	0.71	0.94	0.99
1.9	117	21	4	1376	0.98	0.85	0.97	0.96
1.10	16	13	0	292	0.96	0.55	1.00	0.96
1.11	61	16	3	772	0.98	0.79	0.95	0.98
1.13	16	15	2	315	0.95	0.52	0.89	0.95
1.14	106	41	13	1462	0.97	0.72	0.89	0.95
1.15	54	4	0	582	0.99	0.93	1.00	0.99
1.17	22	6	0	285	0.98	0.79	1.00	0.97
1.18	32	10	1	424	0.98	0.76	0.97	0.98

Table S10: The performance of the ADIOS algorithm versus the SVM-Prot system on level 2.

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). We have calculated the compression at several points along the curves of the ATIS-CFG recall/precision graph (Figure 3B in the main paper). Figure S11 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

class	TP	FN	FP	TN	Q	Recall	Precision
1.1.1	331	67	48	1241	0.93	0.83	0.87
1.1.3	4	4	0	80	0.95	0.50	1.00
1.1.99	6	8	0	147	0.95	0.43	1.00
1.10.2	8	8	1	166	0.95	0.50	0.89
1.10.3	6	3	0	95	0.97	0.67	1.00
1.10.99	3	0	0	30	1.00	1.00	1.00
1.11.1	62	15	4	771	0.98	0.81	0.94
1.12.99	6	0	0	65	1.00	1.00	1.00
1.13.11	15	12	0	277	0.96	0.56	1.00
1.13.12	0	3	0	30	0.91	0.00	0.00
1.14.11	8	3	0	117	0.98	0.73	1.00
1.14.12	4	1	0	55	0.98	0.80	1.00
1.14.13	14	11	1	251	0.96	0.56	0.93
1.14.14	48	9	0	572	0.99	0.84	1.00
1.14.15	8	1	0	95	0.99	0.89	1.00
1.14.16	6	0	0	67	1.00	1.00	1.00
1.14.18	1	2	0	35	0.95	0.33	1.00
1.14.19	6	0	0	65	1.00	1.00	1.00
1.14.99	15	3	0	180	0.98	0.83	1.00
1.15.1	53	5	2	580	0.99	0.91	0.96
1.16.1	2	3	0	52	0.95	0.40	1.00
1.17.4	21	7	1	281	0.97	0.75	0.95
1.18.1	7	4	0	117	0.97	0.64	1.00
1.18.6	25	5	0	307	0.99	0.83	1.00
1.2.1	95	29	4	1236	0.98	0.77	0.96
1.2.3	3	0	0	32	1.00	1.00	1.00
1.2.4	10	6	0	165	0.97	0.63	1.00
1.2.7	2	6	0	82	0.93	0.25	1.00
1.2.99	2	5	0	72	0.94	0.29	1.00
1.21.3	3	0	0	32	1.00	1.00	1.00
1.3.1	29	8	1	369	0.98	0.78	0.97
1.3.3	23	11	0	347	0.97	0.68	1.00
1.3.5	4	0	0	45	1.00	1.00	1.00
1.3.7	0	3	0	37	0.93	0.00	0.00
1.3.99	13	5	1	181	0.97	0.72	0.93
1.4.1	15	5	0	207	0.98	0.75	1.00
1.4.3	10	12	0	222	0.95	0.45	1.00
1.4.4	2	1	0	35	0.97	0.67	1.00
1.4.99	6	1	0	77	0.99	0.86	1.00
1.5.1	18	12	1	299	0.96	0.60	0.95
1.5.3	0	3	0	37	0.93	0.00	0.00
1.5.99	2	1	0	37	0.98	0.67	1.00
1.6.1	4	1	0	52	0.98	0.80	1.00
1.6.2	5	0	0	50	1.00	1.00	1.00
1.6.5	162	5	8	1512	0.99	0.97	0.95
1.6.99	24	19	8	429	0.94	0.56	0.75

Table S11: The performance of the ADIOS algorithm on level 3.

class	TP	FN	FP	TN	Q	Recall	Precision
1.7.1	10	4	0	145	0.97	0.71	1.00
1.7.2	5	0	0	55	1.00	1.00	1.00
1.7.3	3	2	0	50	0.96	0.60	1.00
1.7.99	5	5	0	107	0.96	0.50	1.00
1.8.1	30	4	0	345	0.99	0.88	1.00
1.8.4	30	4	0	342	0.99	0.88	1.00
1.9.3	110	28	6	1374	0.98	0.80	0.95
1.97.1	3	0	0	32	1.00	1.00	1.00

Table S11 (continued): The performance of the ADIOS algorithm on level 3.

Enzyme Class	Pattern
1.1.1	WSG {VNVAGV, RT}
1.1.1	GKVIKCKAA VL
1.1.1	ALVTG {AGK, ST, AAQ, AS, SR, SK, TS, NK} GIG
1.1.1	ANQNGAIWKLDLG LDA
1.1.1	AAY {GEV, SSVL, STV, SSV} {MN,AQA}
1.1.1	LTNKNV IFVAGLGGIGLDTS
1.2.1	IF IDG EH GTTGLQI
1.2.1	VSV IDNLVKGA GQAIQN
1.4.3	TG {FQ,GI} YGL
1.6.5	TD {RVL, LKSLI} AY
1.6.5	IAL {TSL, ME, PT} HT
1.8.1	FT {EL, VLPM, HL} YP
1.8.4	EVR {SAHG,SNA,KNA,RAA,SKL,RFA,KYD} DS
1.8.4	{NR,TT} QG
1.11.1	VKFHWKPTCGVK {SM, CL}
1.11.1	{QE,QP} WWPAD
1.11.1	{AI,AP} KFPDFIHTQKR
1.11.1	FDHER IPERVVHARG
1.11.1	GIPASYR HM GFGSHT
1.11.1	VS LDKARRLLWPIKQKYG
1.15.1	FW {VVN,LVN,MP} WD
1.18.6	{IPL,CIG} VHGGQGC MFV

Table S12: Some of the specific ADIOS patterns appear in specific Enzyme Classes.

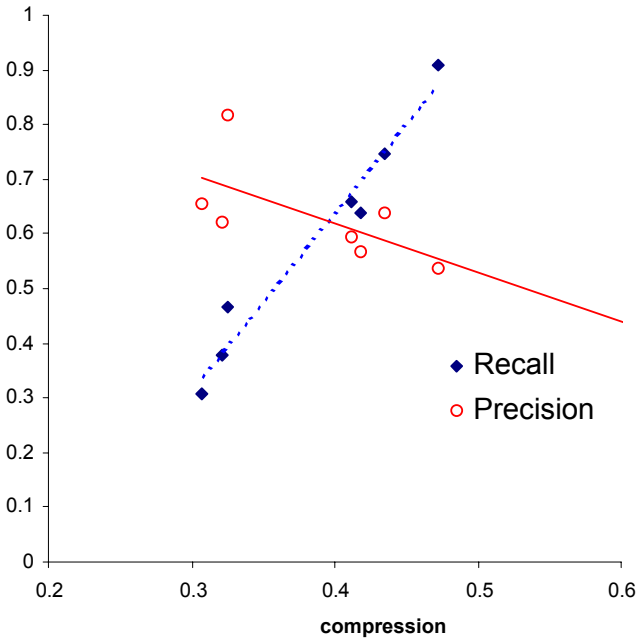

Figure S11: Correlation between the recall/precision levels (ordinate, blue and red respectively), versus compression rate (abscissa), obtained for the ATIS-CFG problem (Figure 3B in the main paper).

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 Figure S12, 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 is consistent with the actual tests described in the main paper.

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 (*12, 13*) and for natural language engineers (*14–16*). Because a completely bias-free unsupervised learning is impossible (*12, 17, 18*), 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 (*19, 20*) rules — "inventories" of units of varying kinds and sizes (*21, 22*) such as: idioms and semi-

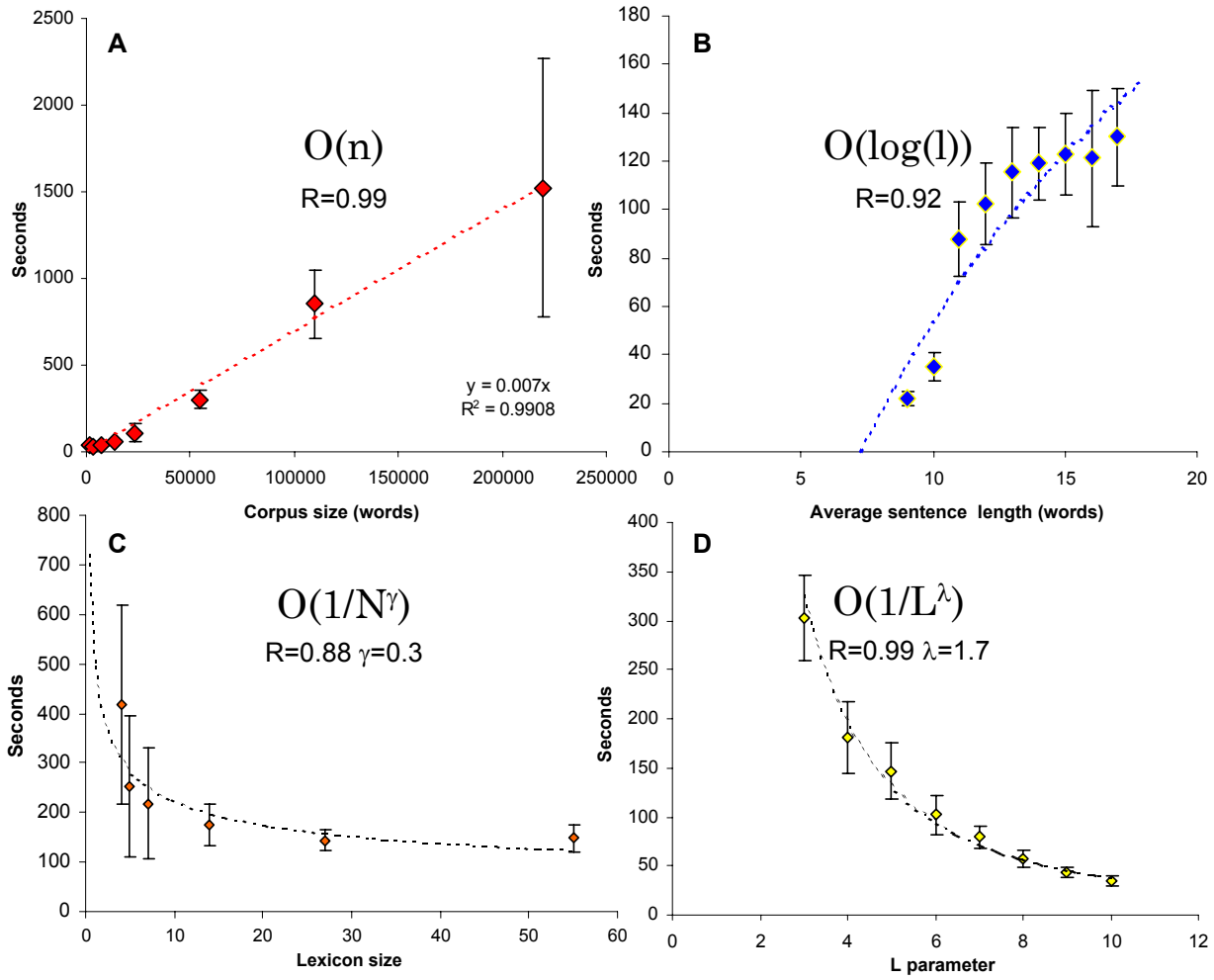

Figure S12: Four experiments estimating the computational complexity of ADIOS by measuring the training time until convergence (ordinate) on the TA1 grammar versus: (A) The total number of words in the corpus; (B) The average sentence length (the experiment manipulated the average sentence length without increasing the total number of words in the corpus); (C) The initial lexicon size; (D) The value of the context parameter L.

productive forms (*23, 24*), prefabricated expressions (*25, 26*), "syntactic nuts" (*27*), frequent collocations (*28*), multiword expressions (*29, 30*), and constructions (*31–34*). 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 (*35*) and semantics-oriented resources such as the WordNet (*36*), the PhraseNet (*37*), and the Berkeley FrameNet (*38, 39*).

References and Notes

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

2. B. Moore, J. Carroll, Parser comparison – context-free grammar (CFG) data (2001). Online at http://www.informatics.susx.ac.uk/research/nlp/carroll/cfg-resources/.

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

4. R. L. Gómez, *Psychological Science* **13**, 431 (2002).

5. B. MacWhinney, C. Snow, *Journal of Computational Lingustics* **12**, 271 (1985).

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

7. M. C. Linebarger, M. Schwartz, E. Saffran, *Cognition* **13**, 361 (1983).

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

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

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

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

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

13. J. L. Elman, *et al.*, *Rethinking innateness: A connectionist perspective on development* (MIT Press, Cambridge, MA, 1996).

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

15. A. Clark, Unsupervised language acquisition: Theory and practice, Ph.D. thesis, COGS, University of Sussex (2001).

16. A. Roberts, E. Atwell, Unsupervised grammar inference systems for natural language, *Tech. Rep. 2002.20*, School of Computing, University of Leeds (2002).

17. M. A. Nowak, N. L. Komarova, P. Niyogi, *Science* **291**, 114 (2001).

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

19. H. Daumé III, K. Knight, I. Langkilde-Geary, D. Marcu, K. Yamada, *Proceedings of the 2002 International Conference on Natural Language Generation (INLG – 2002)* (Harriman, NY, 2002), pp. 9–16.

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

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

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

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

24. B. Erman, B. Warren, *Text* **20**, 29 (2000).

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

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

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

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

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

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

31. P. Kay, C. J. Fillmore, *Language* **75**, 1 (1999).

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

33. A. E. Goldberg, *Trends in Cognitive Sciences* **7**, 219 (2003).

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

35. M. P. Marcus, B. Santorini, M. A. Marcinkiewicz, *Computational Linguistics* **19**, 313 (1994).

36. G. A. Miller, C. Fellbaum, *Cognition* **41**, 197 (1991).

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

38. C. F. Baker, C. J. Fillmore, J. B. Lowe, *Proceedings of the COLING-ACL* (Montreal, Canada, 1998).

39. C. F. Baker, C. J. Fillmore, B. Cronin, *International Journal of Lexicography* **16**, 281 (2003).