Automatic acquisition and efficient representation of syntactic structures

Zach Solan, Eytan Ruppin, David Horn
Faculty of Exact Sciences
Tel Aviv University
Tel Aviv, Israel 69978
{*rsolan,ruppin,horn*}*.post.tau.ac.il*

Shimon Edelman
Department of Psychology
Cornell University
Ithaca, NY 14853, USA
se37@cornell.edu

Abstract

The principle of complementary distributions [1, 2], according to which morphemes that occur in identical contexts belong, in some sense, to the same category, has been advanced as a means for extracting syntactic structures from corpus data. We extend this principle by applying it recursively, and by using mutual information for estimating category coherence. The resulting model learns, in an unsupervised fashion, highly structured, distributed representations of syntactic knowledge from corpora. It also exhibits promising behavior in tasks usually thought to require representations anchored in a grammar, such as systematicity.

1 Motivation

Models dealing with the acquisition of syntactic knowledge are sharply divided into two classes, depending on whether they subscribe to some variant of the classical generative theory of syntax, or operate within the framework of "general-purpose" statistical or distributional learning. An example of the former is the model of [3], which attempts to learn syntactic structures such as Functional Category, as stipulated by the Government and Binding version of generative grammar [4]. An example of the latter model is Elman's widely used Simple Recursive Network (SRN) [5, 6].

We believe that polarization between statistical and classical (generative, rule-based) approaches to syntax is counterproductive, because it hampers the integration of the stronger aspects of each method into a common powerful framework. Indeed, on the one hand, the statistical approach is geared to take advantage of the considerable progress made to date in the areas of distributed representation, probabilistic learning, and "connectionist" modeling. Yet, generic connectionist architectures are ill-suited to the abstraction and processing of symbolic information. On the other hand, classical rule-based systems excel in just those tasks, yet are brittle and difficult to train.

We present a scheme that is tailored to the acquisition of "raw" syntactic information construed in a distributional sense, yet also supports the distillation of rule-like regularities out of the accrued statistical knowledge. Our research is motivated by linguistic theories that combine syntactic transformations with reliance on distributional cues, instead of forgoing one for the other; a good example is the work of Zellig Harris [1, 2].

2 The LKR model

The LKR (Linguistic Knowledge Representation) model constructs syntactic representations of a sample of language from textual corpus data. The model consists of two elements: (1) a Representational Data Structure (RDS) graph, and (2) a Pattern Acquisition (PA) algorithm that learns the RDS in an unsupervised fashion. The PA algorithm aims to detect *patterns* — repetitive sequences of "significant" strings of primitives occurring in the corpus (Figure 1). In that, it is related to prior work on alignment-based learning [7] and regular expression ("local grammar") extraction [8] from corpora. We stress, however, that our algorithm requires no pre-judging either of the scope of the primitives or of their classification, say, into syntactic categories: all the information needed for its operation is extracted from the corpus in an unsupervised fashion.

The initial phase of the PA algorithm involves a preprocessing stage, in which the text is segmented down to the smallest possible morphological constituents (e.g., `ed` is split off both `walked` and `bed`; the algorithm later discovers that `bed` should be left whole, on statistical grounds).[1] The strings of constituents are encoded to form the vertex set of the directed RDS graph, with an edge between two vertices inserted whenever the corresponding transition exists in the corpus (Figure 2, (a)). Thus, corpus sentences initially correspond to paths in the graph.

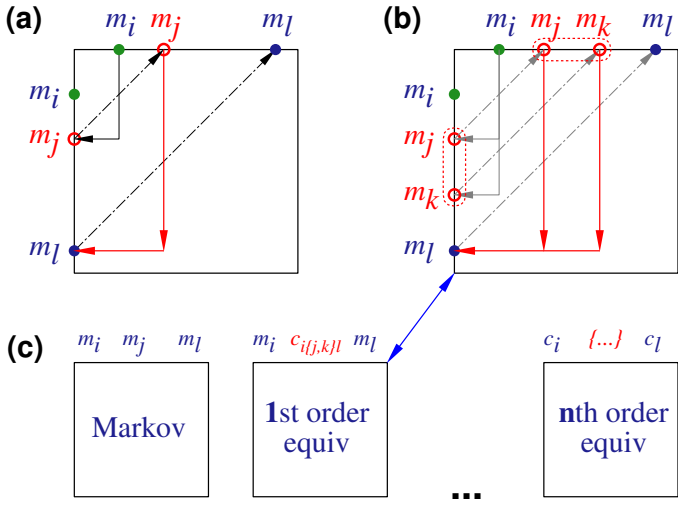

Figure 1: (a) Using the morpheme×morpheme matrix to follow the transitions found in the sequence m_i, m_j, m_l. (b) Two sequences m_i, m_j, m_l and m_i, m_k, m_l form a pattern, which allows m_j and m_k to be attributed to the same equivalence class (following the principle of complementary distributions [1]). The pattern can then serve as a constituent $c_{i\{j,k\}l} \doteq m_i, \{m_j, m_k\}, m_l$ in its own right, and can in turn participate in the formation of higher-order patterns. Recursively abstracting patterns from a corpus allows us to capture the syntactic regularities in a concise, yet highly expressive formalism. (c) The recursive pattern abstraction approach (right) is qualitatively different from (and more powerful than) a simple tabulation of $m_i \rightarrow m_j$ transition probabilities (left).

In the second phase, the PA algorithm makes a pass over the graph and detects Significant Patterns (sequences of constituents) (SP), which are then used to modify the RDS graph (Algorithm 1). The algorithm scans the graph path by path, constructing for each path p_i a

[1] We remark that the algorithm can work in any language, with any set of tokens, including individual characters – or phonemes, if applied to speech.

list of candidate constituents, $c_{i1}, \ldots, c_{ik}$. Each of these consists of the following regular expression: a "prefix" (sequence of graph edges), an equivalence class of vertices, and a "suffix" (another sequence of edges; cf. Figure 2, (b)).

Algorithm 1 PA (pattern acquisition), phase 2

```
1:  while patterns exist do
2:    for all path ∈ graph do {path=sentence; graph=corpus}
3:      for all source_node ∈ path do
4:        for all sink_node ∈ path do {source and sink can be equivalence classes}
5:          degree_of_separation = path_index(sink) − path_index(source);
6:          pattern_table ⇐ detect_patterns(source, sink, degree_of_separation, equiva-
            lence_table);
7:        end for
8:      end for
9:      winner ⇐ get_most_significant_pattern(pattern_table);
10:     equivalence_table ⇐ detect_equivalences(graph, winner);
11:     graph ⇐ rewire_graph(graph, winner);
12:   end for
13: end while
```

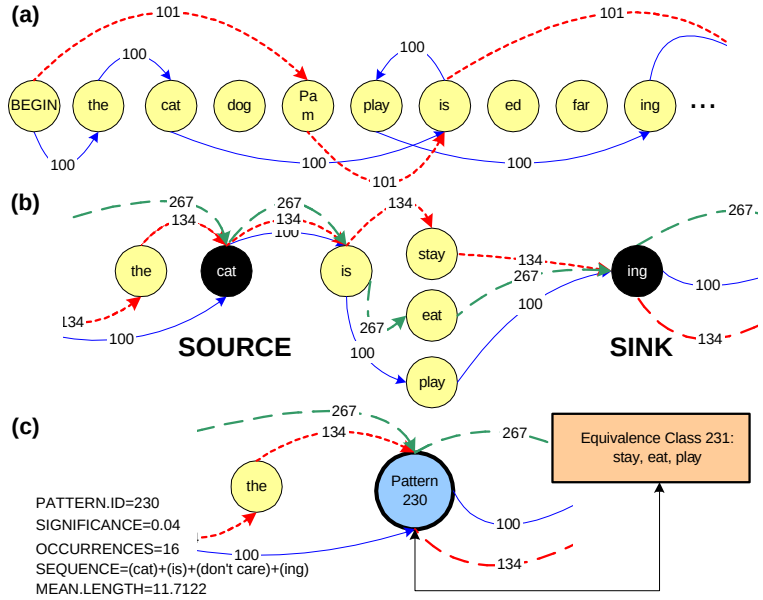

Figure 2: (a) A small portion of the RDS graph for a simple corpus, with sentence #100 (`the cat is play -ing`) indicated by solid arcs. (b) This sentence joins a pattern `cat is {stay, eat, play} -ing`, in which two others (#134,267) already participate. (c) The abstracted pattern and the equivalence class associated with it.

The most significant pattern among the current set of candidates is the one whose constituents $c_1, \ldots, c_k$ have the highest mutual information, defined in the usual manner:

$$I(c_1, c_2, \ldots, c_k) \quad = \quad P(c_1, c_2, \ldots, c_k) \log \frac{P(c_1, c_2, \ldots, c_k)}{\Pi_{j=1}^{k} P(c_j)} \tag{1}$$

Note that the constituents c_j can be morphemes (including "entire" words), or, recursively, equivalence classes and patterns. The probabilities associated with a constituent are esti-

mated from frequencies that are immediately available in the graph (e.g., the out-degree of a node is related to the marginal probability of the corresponding constituent). A pattern tagged as significant is added as a new vertex to the RDS graph, replacing the constituents and edges it subsumes. When an SP is identified, the distinct constituents appearing in the interim slots of any of its occurrences in the graph are collated into an equivalence class associated with it in the graph (Figure 2).

During the pass over the corpus the list of equivalence sets and their composition are updated continuously. At any given stage, the identification of new significant patterns is done using the *current* equivalence sets. Thus, as the algorithm processes more and more text, it "bootstraps" itself and enriches the RDS graph structure with new SPs and their accompanying equivalence sets. The recursive nature of this process enables the algorithm to form more and more complex patterns, in a hierarchical manner. The relationships among these can be visualized recursively in a tree format, with tree depth corresponding to the level of recursion (e.g., Figure 3, (c)). The PA algorithm halts if it processes a given amount of text without finding a new SP or equivalence set (in real-life language acquisition this process may never stop).

The RDS graph acquired in this manner represents syntactic information on a number of levels, all of which use the same basic mechanism of recursive pattern subsumption. Each vertex stands for some SP, i.e., a string of "morphological" constituents that together form a frequent pattern. Such patterns — which eventually can become highly abstract, thus endowing the model with an ability to generalize to unseen inputs — are necessarily "legal" in the sense that their appearance in the corpus is guaranteed. It should be observed that grammaticality here is sanctioned by the corpus, rather than being defined by an *a priori* grammar [2].

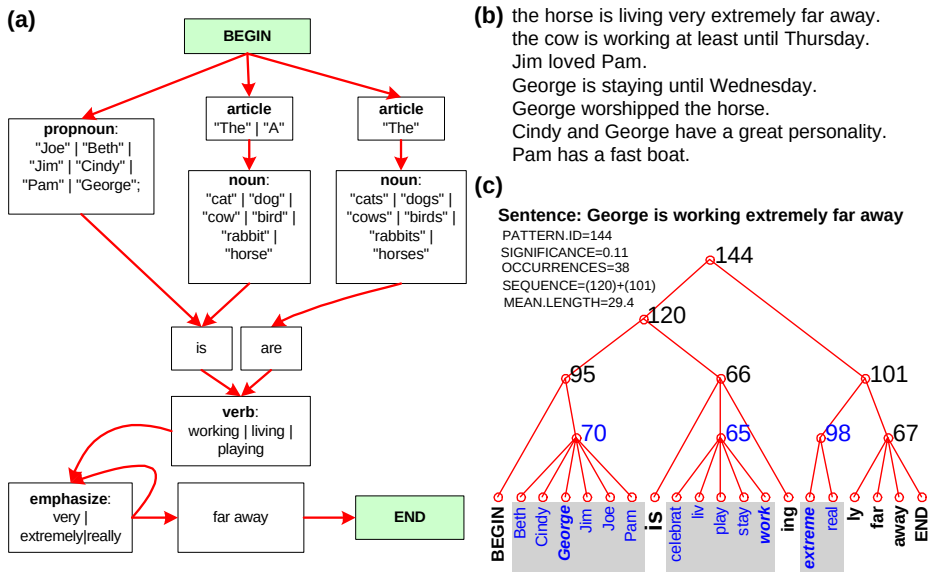

Figure 3: (a) A part of a simple grammar. (b) Some sentences generated by this grammar. (c) The pattern structure of a sample sentence, presented in the form of a tree that captures the hierarchical relationships among constituents. Three equivalence classes are shown explicitly (highlighted).

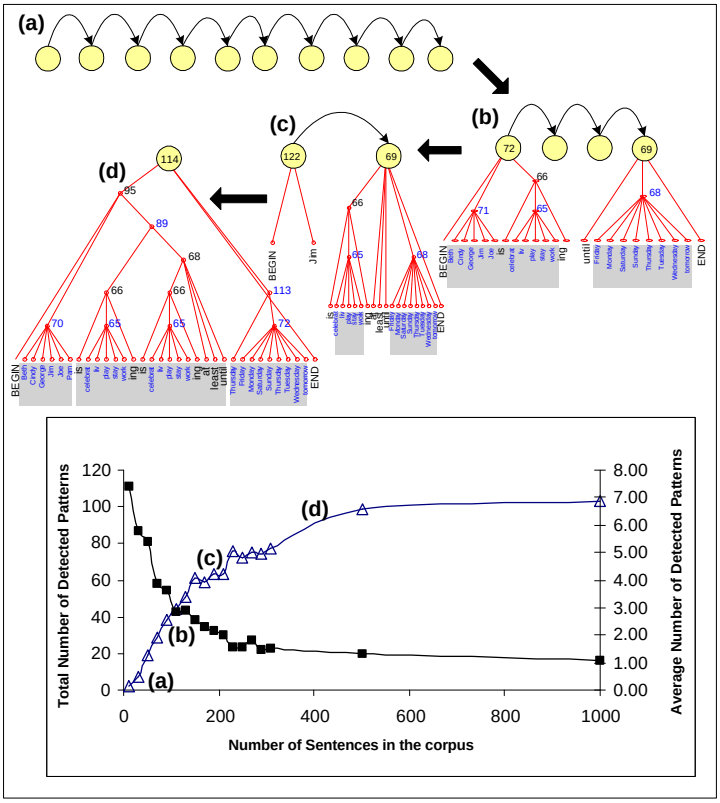

Figure 4: *Top:* the build-up of structured information with progressive exposure to a corpus generated by the simple grammar of Figure 3. (a) Prior to exposure. (b) 100 sentences. (c) 200 sentences. (d) 400 sentences. *Bottom:* the total number of detected patterns (△) and the average number of patterns in a sentence (■), plotted vs. corpus size.

3 Results

We now briefly describe the results of several studies designed to evaluate the viability of the LKR model, in which it was exposed to corpora of varying size and complexity.

Emergence of syntactic structures. Figure 3 shows an example of a sentence from a corpus produced by a simple artificial grammar and its LKR analysis (the use of a simple grammar, constructed with Rmutt, http://www.schneertz.com/rmutt, in these initial experiments allowed us to examine various properties of the model on tightly controlled data). The abstract representation of the sample sentence in Figure 3,(c) looks very much like a parse tree, indicating that our method successfully identified the grammatical structure used to generate its data. To illustrate the gradual emergence of LKR's ability for such concise representation of syntactic structures, we show in Figure 4, top, four trees built for the same sentence after exposing the model to progressively more data from the same corpus. Note that both the number of distinct patterns and the average number of patterns per sentence asymptote for this corpus after exposure to about 500 sentences (Figure 4, bottom).

Scaling to a more realistic corpus. To illustrate the scalability of our method, we describe here briefly the outcome of applying the PA algorithm to a subset of the CHILDES

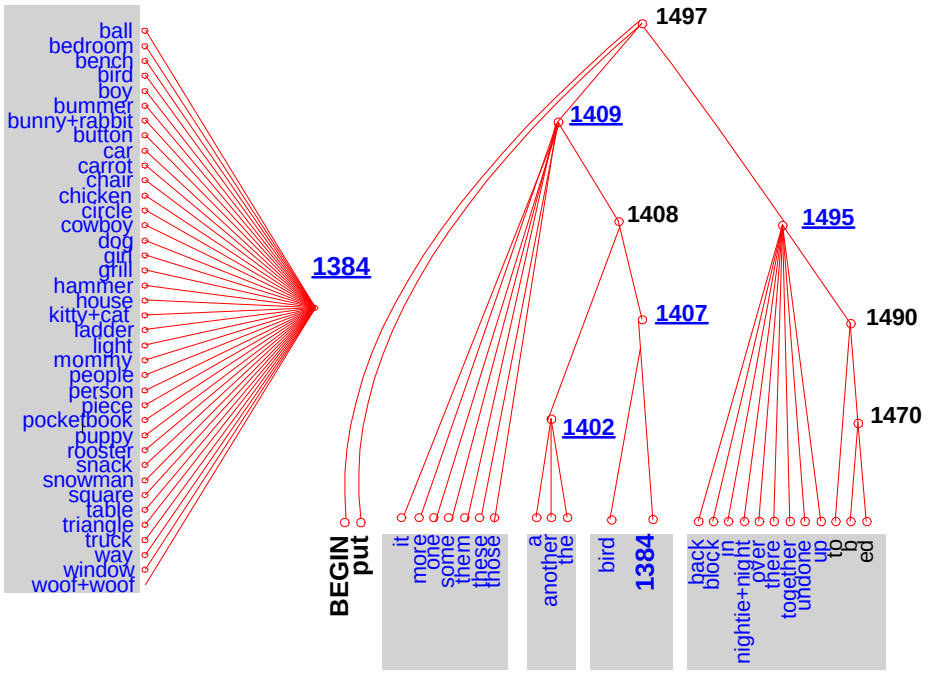

Figure 5: A typical pattern extracted from a subset of the CHILDES corpora collection [9]; equivalence class #1384 is shown on the left, for clarity. Hundreds of such patterns and equivalence classes (underscored in this figure) together constitute a concise representation of the raw data. Although this representational format looks like a collection of finite automata, the information it contains is much richer, because of the recursive invocation of one pattern by another, because of the context sensitivity implied by such connections among patterns (which can and do appear in the company of other patterns in the RDS graph), and because of the global structure imposed by the web of (local) patterns acting together.

collection [9], which consists of transcribed speech produced by, or directed at, children. The corpus we selected contained 9665 sentences (74500 words) produced by parents. The results, one of which is shown in Figure 5, were encouraging: the algorithm found intuitively significant SPs and produced semantically adequate corresponding equivalence sets. Altogether, 1062 patterns and 775 equivalence classes were established. Representing the corpus in terms of these constituents resulted in a significant compression: the average number of constituents per sentence dropped from 6.70 in the raw data to 2.18 after training, and the entropy was reduced from 2.6 to 1.5.

Systematicity. An important characteristic of a cognitive representation scheme is its systematicity, measured by the ability to deal properly with structurally related items (see [10] for a definition and discussion). We have assessed the systematicity of the LKR model by splitting the corpus generated by the grammar of Figure 3 into training and test sets. All three parts of the learning process – detection (line 6 in Algorithm 1), equivalence class extraction (line 10) and graph rewiring (line 11) – were carried out on the training set, resulting in the usual RDS graph structure. In parallel, a separate graph was constructed for the test set, by carrying out only the rewiring steps (line 11). We then examined the representations of sentences from the test set (which could be considered "unseen" because

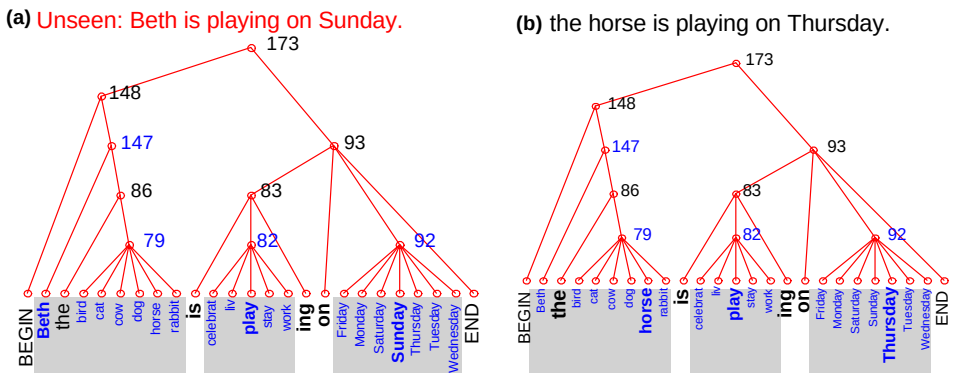

Figure 6: (a) Structured representation of an "unseen" sentence that had been excluded from the corpus used to learn the patterns; note that the detected structure is identical to that of (b), a "seen" sentence. The identity between the structures detected in (a) and (b) is a manifestation of Level-3 systematicity of the LKR model ("Novel Constituent: the test set contain at least one atomic constituent that did not appear anywhere in the training set"; see [10], pp.3-4).

they had no effect on the learning of the test-set graph), and compared their structure to that of similar sentences from the training set. A typical result appears in Figure 6; the general finding was of Level 3 systematicity according to the nomenclature of [10].

Semantics. To show that the graph produced by the PA algorithm carries information about the semantics (and not only the syntax) of the corpus used to train it, we submitted the list of equivalence sets formed by the algorithm to multidimensional scaling. We first embedded the representation into a high-dimensional space by constructing an index matrix of words by equivalence class elements; a value of zero or one is assigned to the matrix entry depending on whether or not the word in question belongs to the equivalence set. A 2D visualization of the resulting semantic space, obtained by reducing the dimensionality of the row space of this matrix, appears in Figure 7.

4 Concluding remarks

We have described a linguistic pattern acquisition algorithm that aims to achieve a streamlined representation (1) by striving to minimize the number of constituents and the total "message" length, (2) by compactly representing recursively structured constituent patterns as single vertices in the graph, and (3) by placing element-wise distinct strings that have an identical backbone and similar syntactic structure into the same equivalence class. In this matter, the PA action follows the Minimum Description Length (MDL) principle [11] (the many interesting parallels between our approach and a theory of perception and learning related to MDL [12] are beyond the scope of the present paper). Finding a good set of constituents leads to the emergence of a stationary representation of language, which eventually changes less and less with progressive exposure to more data. The power of the constituent graph representation stems from the interacting ensembles of local patterns that comprise it. Together, the local constituent patterns create global complexity and impose long-range order on the linguistic structures they encode.

References

[1] Z. S. Harris. Distributional structure. *Word*, 10:140–162, 1954.

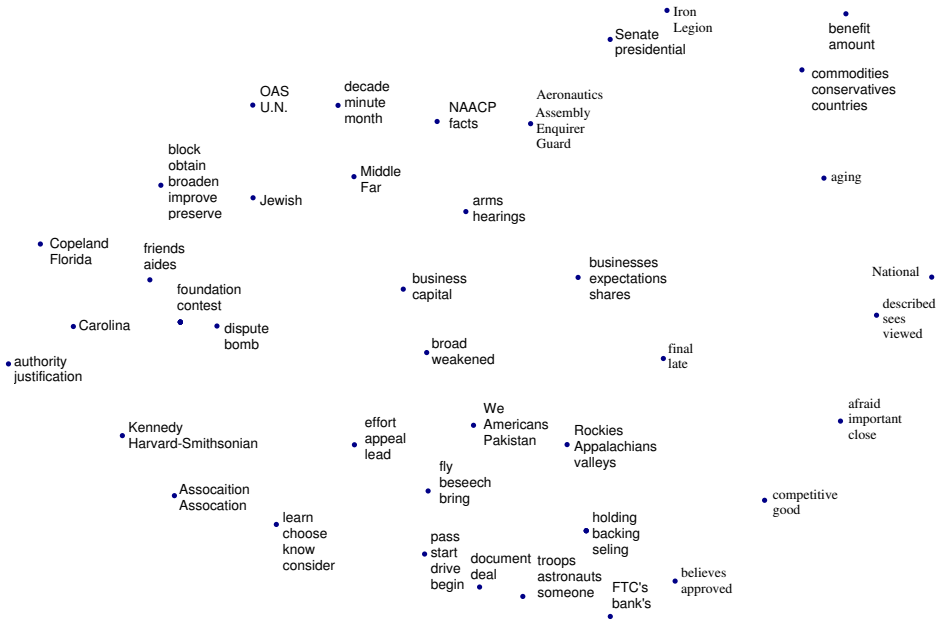

Figure 7: Some of the semantic information captured by the LKR model (the misspellings in this plot are the same as in the corpus used to generate it). Multidimensional scaling was used to visualize an auxiliary graph in which the edges connect any two words that appear in the same RDS class. For the 3D solution (which is shown here projected into 2D), the MDS stress was about 0.3 (the stress decreased to 0.15 for a 5D solution). Labels are shown for some representative words; members of the same cluster tend to be related both semantically and syntactically. The simple visualization method used to generate this figure conflates different senses of the same word.

[2] Z. S. Harris. *A theory of language and information*. Clarendon Press, Oxford, 1991.

[3] R. Kazman. Simulating the child's acquisition of the lexicon and syntax - experiences with Babel. *Machine Learning*, 16:87–120, 1994.

[4] N. Chomsky. *Lectures on Government and Binding*. Foris, Dordrecht, 1981.

[5] J. L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[6] M. H. Christiansen and N. Chater. Connectionist psycholinguistics: Capturing the empirical data. *Trends in Cognitive Sciences*, 5:82–88, 2001.

[7] M. van Zaanen. ABL: Alignment-Based Learning. In *COLING 2000 - Proc. of the 18th International Conference on Computational Linguistics*, p.961–967, 2000.

[8] M. Gross. The construction of local grammars. In E. Roche and Y. Schabès, editors, *Finite-State Language Processing*, p.329–354. MIT Press, Cambridge, MA, 1997.

[9] B. MacWhinney and C. Snow. The child language exchange system. *Journal of Computational Lingustics*, 12:271–296, 1985.

[10] T. J. van Gelder and L. Niklasson. On being systematically connectionist. *Mind and Language*, 9:288–302, 1994.

[11] J. Rissanen. Minimum description length principle. In S. Kotz and N. L. Johnson, editors, *Encyclopedia of Statistic Sciences*, volume 5, p.523–527. Wiley, 1987.

[12] H. B. Barlow. Conditions for versatile learning, Helmholtz's unconscious inference, and the task of perception. *Vision Research*, 30:1561–1571, 1990.