Learning Auxiliary Fronting with Grammatical Inference

Alexander Clark (alexc@cs.rhul.ac.uk)
Department of Computer Science
Royal Holloway University of London
Egham, Surrey TW20
Rémi Eyraud (remi.eyraud@univ-st-etienne.fr)
EURISE
23, rue du Docteur Paul Michelon
42023 Saint-Étienne Cedex 2
France

Abstract

We present a simple context-free grammatical inference algorithm, and prove that it is capable of learning an interesting subclass of context-free languages. We also demonstrate that an implementation of this algorithm is capable of learning auxiliary fronting in polar interrogatives (AFIPI) in English. This has been one of the most important test cases in language acquisition over the last few decades. We demonstrate that learning can proceed even in the complete absence of examples of particular constructions, and thus that debates about the frequency of occurrence of such constructions are irrelevant. We discuss the implications of this on the type of innate learning biases that must be hypothesized to explain first language acquisition.

Introduction

For some years, a particular set of examples has been used to provide support for nativist theories of first language acquisition (FLA). These examples, which hinge around auxiliary inversion in the formation of questions in English, have been considered to provide a strong argument in favour of the nativist claim: that FLA proceeds primarily through innately specified domain specific mechanisms or knowledge, rather than through the operation of general-purpose cognitive mechanisms. A key point of empirical debate is the frequency of occurrence of the forms in question. If these are vanishingly rare, or non-existent in the primary linguistic data, and yet children acquire the construction in question, then the hypothesis that they have innate knowledge would be supported. But this rests on the assumption that examples of that specific construction are necessary for learning to proceed. In this paper we show that this assumption is false: that this particular construction can be learned without the learner being exposed to any examples of that particular type. Our demonstration is primarily mathematical/computational: we present a simple experiment that demonstrates the applicability of this approach to this particular problem neatly, but the data we use is not intended to be a realistic representation of the primary linguistic data, nor is the particular algorithm we use suitable for large scale grammar induction.

We present a general purpose context-free grammatical algorithm that is provably correct under a certain learning criterion. This algorithm incorporates no domain specific knowledge: it has no specific information about language; no knowledge of X-bar schemas, no hidden sources of information to reveal the structure. It operates purely on unannotated strings of raw text. Obviously, as all learning algorithms do, it has an implicit learning bias. This very simple algorithm has a particularly clear bias, with a simple mathematical description, that allows a remarkably simple characterisation of the set of languages that it can learn. This algorithm does not use a statistical learning paradigm that has to be tested on large quantities of data. Rather it uses a symbolic learning paradigm, that works efficiently with very small quantities of data, while being very sensitive to noise. We discuss this choice in some depth below.

For reasons that were first pointed out by Chomsky [Chomsky, 1975, pages 129–137], algorithms of this type are not capable of learning all of natural language. It turns out, however, that algorithms based on this approach are sufficiently strong to learn some key properties of language, such as the correct rule for forming polar questions.

In the next section we shall describe the dispute briefly; in the subsequent sections we will describe the algorithm we use, and the experiments we have performed.

The Dispute

We will present the dispute in traditional terms, though later we shall analyse some of the assumptions implicit in this description. In English, polar interrogatives (yes/no questions) are formed by fronting an auxiliary, and adding a dummy auxiliary "do" if the main verb is not an auxiliary. For example,

Example 1a The man is hungry.

Example 1b Is the man hungry?

When the subject NP has a relative clause that also contains an auxiliary, the auxiliary that is moved is not the auxiliary in the relative clause, but the one in the main (matrix) clause.

Example 2a The man who is eating is hungry.

Example 2b Is the man who is eating hungry?

An alternative rule would be to move the first occurring auxiliary, i.e. the one in the relative clause, which would produce the form

Example 2c Is the man who eating is hungry?

In some sense, there is no reason that children should favour the correct rule, rather than the incorrect one, since they are both of similar complexity and so on. Yet children do in fact, when provided with the appropriate context, produce sentences of the form of Example 2b, and rarely if ever produce errors of the form Example 2c. [Crain and Nakayama, 1987] The problem is how to account for this phenomenon.

Chomsky claimed first, that sentences of the type in Example 2b are vanishingly rare in the linguistic environment that children are exposed to, yet when tested they unfailingly produce the correct form rather than the incorrect Example 2c. This is put forward as strong evidence in favour of innately specified language specific knowledge: we shall refer to this view as linguistic nativism.

In a special volume of the Linguistic Review, Pullum and Scholz [Pullum and Scholz, 2002], showed that in fact sentences of this type are not rare at all. Much discussion ensued on this *empirical* question and the consequences of this in the context of arguments for linguistic nativism. These debates revolved around both the methodology employed in the study, and also the consequences of such claims for nativist theories. It is fair to say that in spite of the strength of Pullum and Scholz's arguments, nativists remained completely unconvinced by the overall argument.

[Reali and Christiansen, 2004] present a possible solution to this problem. They claim that local statistics, effectively n-grams, can be sufficient to indicate to the learner which alternative should be preferred. However this argument has been carefully rebutted by [Kam et al., 2005], who show that this argument relies purely on a phonological coincidence in English. This is unsurprising since it is implausible that a flat, finite-state model should be powerful enough to model a phenomenon that is clearly structure dependent in this way.

In this paper we argue that the discussion about the rarity of sentences that exhibit this particular structure is irrelevant: we show that simple grammatical inference algorithms can learn this property even in the complete absence of sentences of this particular type. Thus the issue as to how frequently an infant child will see them is a moot point.

Algorithm

Context-free grammatical inference algorithms are explored in two different communities: in grammatical inference and in NLP. The task in NLP is normally taken to be one of recovering appropriate annotations [Smith and Eisner, 2005] that normally represent constituent structure (strong learning), while in grammatical inference, researchers are more interested in merely identifying the language (weak learning). In both communities, the best performing algorithms that learn from raw positive data only [1], , generally rely on some combination of three heuristics: frequency, information theoretic measures of constituency, and finally substitutability. [2] The first rests on the observation that strings of words generated by constituents are likely to occur more frequently than by chance. The second heuristic looks for information theoretic measures that may predict boundaries, such as drops in conditional entropy.. The third method which is the foundation of the algorithm we use, is based on the distributional analysis of Harris [Harris, 1954]. This principle has been appealed to by many researchers in the field of grammatical inference, but these appeals have normally been informal and heuristic [van Zaanen, 2000].

In its crudest form we can define it as follows: given two sentences "I saw a cat over there", and "I saw a dog over there" the learner will hypothesize that "cat" and "dog" are similar, since they appear in the same context "I saw a __ there". Pairs of sentences of this form can be taken as evidence that two words, or strings of words are substitutable.

Preliminaries

We briefly define some notation.

An *alphabet* Σ is a finite nonempty set of symbols called *letters*. A *string* w over Σ is a finite sequence $w = a_1 a_2 \dots a_n$ of letters. Let $|w|$ denote the length of w. In the following, letters will be indicated by $a, b, c, \dots$, strings by $u, v, \dots, z$, and the empty string by λ. Let Σ^* be the set of all strings, the free monoid generated by Σ. By a language we mean any subset $L \subseteq \Sigma^*$. The set of all substrings of a language L is denoted $Sub(L) = \{u \in \Sigma^+ : \exists l, r, lur \in L\}$ (notice that the empty word does not belong to $Sub(L)$). We shall assume an order $\prec$ or $\preceq$ on Σ which we shall extend to Σ^* in the normal way by saying that $u \prec v$ if $|u| < |v|$ or $|u| = |v|$ and u is lexicographically before v.

A grammar is a quadruple $G = \langle V, \Sigma, P, S \rangle$ where Σ is a finite alphabet of *terminal symbols*, V is a finite alphabet of *variables* or *non-terminals*, P is a finite set of *production rules*, and $S \in V$ is a start symbol.

If $P \subseteq V \times (\Sigma \cup V)^+$ then the grammar is said to be context-free (CF), and we will write the productions as $T \to w$.

We will write $uTv \Rightarrow uwv$ when $T \to w \in P$. $\overset{*}{\Rightarrow}$ is the reflexive and transitive closure of $\Rightarrow$.

In general, the definition of a class $\mathcal{L}$ relies on a class $\mathcal{R}$ of abstract machines, here called *representations*, together with a function $\mathcal{L}$ from representations to languages, that characterize all and only the languages of $\mathcal{L}$: (1) $\forall R \in \mathcal{R}, \mathcal{L}(R) \in \mathcal{L}$ and (2) $\forall L \in \mathcal{L}, \exists R \in \mathcal{R}$ such that $\mathcal{L}(R) = L$. Two representations R_1 and R_2 are *equivalent iff* $\mathcal{L}(R_1) = \mathcal{L}(R_2)$.

Learning

We now define our learning criterion. This is identification in the limit from positive text [Gold, 1967], with polynomial bounds on data and computation, but not on errors of prediction [de la Higuera, 1997].

[1] We do not consider in this paper the complex and contentious issues around negative data.

[2] For completeness we should include lexical dependencies or attraction.

A learning algorithm A for a class of representations $\mathcal{R}$, is an algorithm that computes a function from a finite sequence of strings $s_1, \ldots, s_n$ to $\mathcal{R}$. We define a presentation of a language L to be an infinite sequence of elements of L such that every element of L occurs at least once. Given a presentation, we can consider the sequence of hypotheses that the algorithm produces, writing $R_n = A(s_1, \ldots s_n)$ for the nth such hypothesis.

The algorithm A is said to identify the class $\mathcal{R}$ in the limit if for every $R \in \mathcal{R}$, for every presentation of $\mathcal{L}(R)$, there is an N such that for all $n > N$, $R_n = R_N$ and $\mathcal{L}(R) = \mathcal{L}(R_N)$.

We further require that the algorithm needs only polynomially bounded amounts of data and computation. We use the slightly weaker notion defined by de la Higuera [de la Higuera, 1997].

Definition A representation class $\mathcal{R}$ is identifiable in the limit from positive data with polynomial time and data iff there exist two polynomials $p(), q()$ and an algorithm A such that

1. Given a positive sample S of size m A returns a representation $R \in \mathcal{R}$ in time $p(m)$,
2. For each representation R of size n there exists a characteristic set CS of size less than $q(n)$ such that if $CS \subseteq S$, A returns a representation R' such that $\mathcal{L}(R) = \mathcal{L}(R')$.

Distributional learning

The key to the Harris approach for learning a language L, is to look at pairs of strings u and v and to see whether they occur in the same contexts; that is to say, to look for pairs of strings of the form lur and lvr that are both in L. This can be taken as evidence that there is a non-terminal symbol that generates both strings. In the informal descriptions of this that appear in Harris's work, there is an ambiguity between two ideas. The first is that they should appear in *all* the same contexts; and the second is that they should appear in *some* of the same contexts. We can write the first criterion as follows:

$$\forall l, r \; lur \in L \text{ if and only if } lvr \in L \tag{1}$$

This has also been known in language theory by the name syntactic congruence, and can be written $u \equiv_L v$.

The second, weaker, criterion is

$$\exists l, r \; lur \in L \text{ and } lvr \in L \tag{2}$$

We call this *weak substitutability* and write it as $u \doteq_L v$. Clearly $u \equiv_L v$ implies $u \doteq_L v$ when u is a substring of the language. Any two strings that do not occur as substrings of the language are obviously syntactically congruent but not weakly substitutable.

First of all, observe that syntactic congruence is a purely language theoretic notion that makes no reference to the grammatical representation of the language, but only to the set of strings that occur in it. However there is an obvious problem: syntactic congruence tells us something very useful about the language, but all we can observe is substitutability.

When working within a Gold-style identification in the limit (IIL) paradigm, we cannot rely on statistical properties of the input sample, since they will in general not be generated by random draws from a fixed distribution. This, as is well known, severely limits the class of languages that can be learned under this paradigm. However, the comparative simplicity of the IIL paradigm in the form when there are polynomial constraints on size of characteristic sets and computation[de la Higuera, 1997] makes it a suitable starting point for analysis.

Given these restrictions, One solution to this problem is simply to *define* a class of languages where substitutability implies congruence. We call these the substitutable languages: A language L is substitutable if and only if for every pair of strings u, v, $u \doteq_L v$ implies $u \equiv_L v$. This rather radical solution clearly rules out the syntax of natural languages, at least if we consider them as strings of raw words, rather than as strings of lexical or syntactic categories. Lexical ambiguity alone violates this requirement: consider the sentences "The rose died", "The cat died" and "The cat rose from its basket". A more serious problem is pairs of sentences like "John is hungry" and "John is running", where it is not ambiguity in the syntactic category of the word that causes the problem, but rather ambiguity in the context. Using this assumption, whether it is true or false, we can then construct a simple algorithm for grammatical inference, based purely on the idea that whenever we find a pair of strings that are weakly substitutable, we can generalise the hypothesized language so that they are syntactically congruent.

The algorithm proceeds by constructing a graph where every substring in the sample defines a node. An arc is drawn between two nodes if and only if the two nodes are weakly substitutable with respect to the sample, i.e. there is an arc between u and v if and only if we have observed in the sample strings of the form lur and lvr. Clearly all of the strings in the sample will form a clique in this graph (consider when l and r are both empty strings). The connected components of this graph can be computed in time polynomial in the total size of the sample. If the language is substitutable then these components will correspond to the congruence classes of the language.

There are two ways of doing this: one way, which is perhaps the purest involves defining a reduction system or semi-Thue system which directly captures this generalisation process. The second way, which we present here, will be more familiar to computational linguists, and involves constructing a grammar.

Grammar construction

Simply knowing the syntactic congruence might not appear to be enough to learn a context-free grammar, but in fact it is. In fact given the syntactic congruence, and a sample of the language, we can simple write down a grammar in Chomsky normal form, and under quite weak assumptions this grammar will converge to a cor-

rect grammar for the language.

This construction relies on a simple property of the syntactic congruence, namely that is in fact a congruence: i.e.,

$$u \equiv_L v \text{ implies } \forall l, r \;\; lur \equiv_L lvr$$

We define the syntactic monoid to be the quotient of the monoid $\Sigma^*/\equiv_L$. The monoid operation $[u][v] = [uv]$ is well defined since if $u \equiv_L u'$ and $v \equiv_L v'$ then $uv \equiv_L u'v'$.

We can construct a grammar in the following trivial way, from a sample of strings where we are given the syntactic congruence.

- The non-terminals of the grammar are identified with the congruence classes of the language.
- For any string $w = uv$, we add a production $[w] \rightarrow [u][v]$.
- For all strings a of length one (i.e. letters of Σ), we add productions of the form $[a] \rightarrow a$.
- The sentence symbol is the congruence class which contains all the strings of the language.

This defines a grammar in CNF. At first sight, this construction might appear to be completely vacuous, and not to define any strings beyond those in the sample. The situation where it generalises is when two different strings are congruent: if $uv = w \equiv w' = u'v'$ then we will have two different rules $[w] \rightarrow [u][v]$ and $[w] \rightarrow [u'][v']$, since $[w]$ is the same non-terminal as $[w']$.

A striking feature of this algorithm is that it makes no attempt to identify which of these congruence classes correspond to non-terminals in the target grammar. Indeed that is to some extent an ill-posed question. There are many different ways of assigning constituent structure to sentences, and indeed some reputable theories of syntax, such as dependency grammars, dispense with the notion of constituent structure all together. De facto standards, such as the Penn treebank annotations are a somewhat arbitrary compromise among many different possible analyses. This algorithm instead relies on the syntactic monoid, which expresses the combinatorial structure of the language in its purest form.

Proof

We will now present our main result, with an outline proof. For a full proof the reader is referred to [Clark and Eyraud, 2005].

Theorem 1 This algorithm polynomially identifies in the limit the class of substitutable context-free languages.

Proof (Sketch) We can assume without loss of generality that the target grammar is in Chomsky normal form. We first define a characteristic set, that is to say a set of strings such that whenever the sample includes the characteristic set, the algorithm will output a correct grammar.

We define $w(\alpha) \in \Sigma^*$ to be the smallest word, according to $\prec$, generated by $\alpha \in (\Sigma \cup V)^+$. For each non-terminal $N \in V$ define $c(N)$ to be the smallest pair of terminal strings (l, r) (extending $\prec$ from Σ^* to $\Sigma^* \times \Sigma^*$, in some way), such that $S \stackrel{*}{\Rightarrow} lNr$.

We can now define the characteristic set $CS = \{lwr | (N \rightarrow \alpha) \in P, (l, r) = c(N), w = w(\alpha)\}$. The cardinality of this set is at most $|P|$ which is clearly polynomially bounded. We observe that the computations involved can all be polynomially bounded in the total size of the sample.

We next show that whenever the algorithm encounters a sample that includes this characteristic set, it outputs the right grammar. We write $\hat{G}$ for the learned grammar. Suppose $[u] \stackrel{*}{\Rightarrow}_{\hat{G}} v$. Then we can see that $u \equiv_L v$ by induction on the maximum length of the derivation of v. At each step we must use some rule $[u'] \Rightarrow [v'][w']$. It is easy to see that every rule of this type preserves the syntactic congruence of the left and right sides of the rules. Intuitively, the algorithm will never generate too large a language, since the languages are substitutable. Conversely, if we have a derivation of a string u with respect to the target grammar G, by construction of the characteristic set, we will have, for every production $L \rightarrow MN$ in the target grammar, a production in the hypothesized grammar of the form $[w(L)] \rightarrow [w(M)][w(N)]$, and for every production of the form $L \rightarrow a$ we have a production $[w(L)] \rightarrow a$. A simple resursive argument shows that the hypothesized grammar will generate all the strings in the target language. Thus the grammar will generate all and only the strings required (QED).

Related work

This is the first provably correct and efficient grammatical inference algorithm for a linguistically interesting class of context-free grammars (but see for example [Yokomori, 2003] on the class of very simple grammars). It can also be compared to Angluin's famous work on reversible grammars [Angluin, 1982] which inspired a similar paper[Pilato and Berwick, 1985].

Experiments

We decided to see whether this algorithm without modification could shed some light on the debate discussed above. The experiments we present here are not intended to be an exhaustive testing of the learnability of natural language. The focus is on determining whether learning can proceed in the absence of positive samples, and given only a very weak general purpose bias.

Implementation

We have implemented the algorithm described above. There are a number of algorithmic issues that were addressed. First, in order to find which pairs of strings are substitutable, the naive approach would be to compare strings pairwise which would be quadratic in the

the man who is hungry died .
the man ordered dinner .
the man died .
the man is hungry .
is the man hungry ?
the man is ordering dinner .
is the man who is hungry ordering dinner ?
∗is the man who hungry is ordering dinner ?

Table 1: Auxiliary fronting data set. Examples above the line were presented to the algorithm during the training phase, and it was tested on examples below the line.

number of sentences. A more efficient approach maintains a hashtable mapping from contexts to congruence classes. Caching hashcodes, and using a union-find algorithm for merging classes allows an algorithm that is effectively linear in the number of sentences.

In order to handle large data sets with thousands of sentences, it was necessary to modify the algorithm in various ways which slightly altered its formal properties. However for the experiments reported here we used a version which performs exactly in line with the mathematical description above.

Data

For clarity of exposition, we have used extremely small artificial data-sets, consisting only of sentences of types that would indubitably occur in the linguistic experience of a child.

Our first experiments were intended to determine whether the algorithm could determine the correct form of a polar question when the noun phrase had a relative clause, even when the algorithm was not exposed to any examples of that sort of sentence. We accordingly prepared a small data set shown in Table 1. Above the line is the training data that the algorithm was trained on. It was then tested on all of the sentences, including the ones below the line. By construction the algorithm would generate all sentences it has already seen, so it scores correctly on those. The learned grammar also correctly generated the correct form and did not generate the final form.

We can see how this happens quite easily since the simple nature of the algorithm allows a straightforward analysis. We can see that in the learned grammar "the man" will be congruent to "the man who is hungry", since there is a pair of sentences which differ only by this. Similarly, "hungry" will be congruent to "ordering dinner". Thus the sentence "is the man hungry ?" which is in the language, will be congruent to the correct sentence.

Our second data set is shown in Table 2, and is a fragment of the English auxiliary system. This has also been claimed to be evidence in favour of nativism. This was discussed in some detail by [Pilato and Berwick, 1985]. Again the algorithm correctly learns.

it rains
it may rain
it may have rained
it may be raining
it has rained
it has been raining
it is raining
it may have been raining
∗it may have been rained
∗it may been have rain
∗it may have been rain

Table 2: English auxiliary data. Training data above the line, and testing data below.

Discussion

Chomsky was among the first to point out the limitations of Harris's approach, and it is certainly true that the grammars produced from these toy examples overgenerate radically. On more realistic language samples this algorithm would eventually start to generate even the incorrect forms of polar questions.

Given the solution we propose it is worth looking again and examining why nativists have felt that AFIPI was such an important issue. It appears that there are several different areas. First, the debate has always focussed on how to construct the interrogative from the declarative form. The problem has been cast as finding which auxilary should be "moved". Implicit in this is the assumption that the interrogative structure must be defined with reference to the declarative, one of the central assumptions of traditional transformational grammar. Now, of course, given our knowledge of many different formalisms which can correctly generate these forms without movement we can see that this assumption is false. There is of course a relation between these two sentences, a semantic one, but this does not imply that there need be any particular syntactic relation, and certainly not a "generative" relation.

Secondly, the view of learning algorithms is very narrow. It is considered that only sentences of that exact type could be relevant. We have demonstrated, if nothing else, that that view is false. The distinction can be learnt from a set of data that does not include any example of the exact piece of data required: as long as the various parts can be learned separately, the combination will function in the natural way.

A more interesting question is the extent to which the biases implicit in the learning algorithm are domain specific. Clearly the algorithm has a strong bias. It overgeneralises massively. One of the advantages of the algorithm for the purposes of this paper is that its triviality allows a remarkably clear and explicit statement of its bias. But is this bias specific to the domain of language? It in no way refers to anything specific to the field of language, still less specific to human language – no references to parts of speech, or phrases, or even hierarchical phrase structure. It is now widely recognised

that this sort of recursive structure is domain-general [Jackendoff and Pinker, 2005].

We have selected for this demonstration an algorithm from grammatical inference. A number of statistical models have been proposed over the last few years by researchers such as [Klein and Manning, 2002, Klein and Manning, 2004] and [Solan et al., 2005]. These models impressively manage to extract significant structure from raw data. However, for our purposes, neither of these models is suitable. Klein and Manning's model uses a variety of different cues, which combine with some specific initialisation and smoothing, and an explicit constraint to produce binary branching trees. Though very impressive, the model is replete with domain-specific biases and assumptions. Moreover, it does not learn a language in the strict sense (a subset of the set of all strings), though it would be a simple modification to make it perform such a task. The model by Solan et al. would be more suitable for this task, but again the complexity of the algorithm, which has numerous components and heuristics, and the lack of a theoretical justification for these heuristics again makes the task of identifying exactly what these biases are, and more importantly how domain specific they are, a very significant problem.

Conclusion

We have presented a simple resolution of the argument that the acquisition of auxiliary fronting in polar interrogatives supports linguistic nativism. We have shown how a very simple algorithm based on the ideas of Zellig Harris, can explain this problem with a simple domain-general heuristic. We show that the empirical question as to the frequency of occurrence of polar questions of a certain type in child-directed speech is a moot point, since the distinction in question can be learned even when no such sentences occur.

References

[Angluin, 1982] Angluin, D. (1982). Inference of reversible languages. *Communications of the ACM*, 29:741–765.

[Chomsky, 1975] Chomsky, N. (1975). *The Logical Structure of Linguistic Theory*. University of Chicago Press.

[Clark and Eyraud, 2005] Clark, A. and Eyraud, R. (2005). Identification in the limit of substitutable context free languages. In Jain, S., Simon, H. U., and Tomita, E., editors, *Proceedings of The 16th International Conference on Algorithmic Learning Theory*, pages 283–296. Springer-Verlag.

[Crain and Nakayama, 1987] Crain, S. and Nakayama, M. (1987). Structure dependence in grammar formation. *Language*, 63(522-543).

[de la Higuera, 1997] de la Higuera, C. (1997). Characteristic sets for polynomial grammatical inference. *Machine Learning*, (27):125–138. Kluwer Academic Publishers. Manufactured in Netherland.

[Gold, 1967] Gold, E. M. (1967). Language indentification in the limit. *Information and control*, 10(5):447 – 474.

[Harris, 1954] Harris, Z. (1954). Distributional structure. *Word*, 10(2-3):146–62.

[Jackendoff and Pinker, 2005] Jackendoff, R. and Pinker, S. (2005). The nature of the language faculty and its implications for the evolution of language. *Cognition*, 97:211–225.

[Kam et al., 2005] Kam, X. N. C., Stoyneshka, I., Tornyova, L., Fodor, J. D., and Sakas, W. G. (2005). Non-robustness of syntax acquisition from n-grams: A cross-linguistic perspective. In *The 18th Annual CUNY Sentence Processing Conference*.

[Klein and Manning, 2004] Klein, D. and Manning, C. (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the ACL*.

[Klein and Manning, 2002] Klein, D. and Manning, C. D. (2002). A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the ACL*.

[Pilato and Berwick, 1985] Pilato, S. F. and Berwick, R. C. (1985). Reversible automata and induction of the english auxiliary system. In *Proceedings of the ACL*, pages 70–75.

[Pullum and Scholz, 2002] Pullum, G. K. and Scholz, B. C. (2002). Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, 19(1-2):9–50.

[Reali and Christiansen, 2004] Reali, F. and Christiansen, M. H. (2004). Structure dependence in language acquisition: Uncovering the statistical richness of the stimulus. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society*, Mahwah, NJ. Lawrence Erlbaum.

[Smith and Eisner, 2005] Smith, N. A. and Eisner, J. (2005). Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 354–362, Ann Arbor, Michigan.

[Solan et al., 2005] Solan, Z., Horn, D., Ruppin, E., and Edelman, S. (2005). Unsupervised learning of natural languages. *Proc. Natl. Acad. Sci.*, 102:11629–11634.

[van Zaanen, 2000] van Zaanen, M. (2000). ABL: Alignment-based learning. In *COLING 2000 - Proceedings of the 18th International Conference on Computational Linguistics*.

[Yokomori, 2003] Yokomori, T. (2003). Polynomial-time identification of very simple grammars from positive data. *Theoretical Computer Science*, 298(1):179–206.