Computational Learning of Syntax

Alexander Clark

Department of Philosophy, King's College London, London WC2R 2LS, United Kingdom;
email: alexsclark@gmail.com

Annu. Rev. Linguist. 2017. 3:107–23

First published online as a Review in Advance on September 7, 2016

The *Annual Review of Linguistics* is online at linguist.annualreviews.org

This article's doi:
10.1146/annurev-linguistics-011516-034008

Keywords

syntax, language acquisition, computational modeling

Abstract

Learnability has traditionally been considered to be a crucial constraint on theoretical syntax; however, the issues involved have been poorly understood, partly as a result of the lack of simple learning algorithms for various types of formal grammars. Here I discuss the computational issues involved in learning hierarchically structured grammars from strings of symbols alone. The methods involved are based on an abstract notion of the derivational context of a syntactic category, which in the most elementary case of context-free grammars leads to learning algorithms based on a form of traditional distributional analysis. Crucially, these techniques can be extended to work with mildly context-sensitive grammars (and beyond), thus leading to learning methods that can in principle learn classes of grammars that are powerful enough to represent all natural languages. These learning methods require that the syntactic categories of the grammars be visible in a certain technical sense: They must be well characterized either by the sets of symbols that they generate or by the sets of contexts in which they can appear. However, there are still significant gaps between these theoretical results and their direct implementation as models of language acquisition; I discuss how these remaining problems can be overcome.

1. INTRODUCTION

In this article I review a recently developed class of algorithms for the computational learning of various families of formal grammars. These algorithms take as input sequences of symbols, or strings, from a language, and sometimes information about what strings are not in the language, and output grammars of various types; crucially, they come with mathematical proofs of their correctness. Although there are a wide range of these algorithms, which operate under different assumptions about the information available and produce a variety of different types of grammars, they are all based on the same abstract principles. Every class of grammars has a particular derivation process, which takes parts of a derivation and combines them into larger units. By modeling the relationship between the derivation yields and the derivation contexts—the process that turns small units into complete sentences—we can construct a grammar that is, under certain assumptions, guaranteed to be correct. For the first time, we have a family of simple algorithms based on sound mathematical principles that are capable of learning large classes of grammars. The largest classes of grammars are extremely powerful: Even under rather pessimistic assumptions, they contain grammars that are powerful enough to represent natural languages.

The goal of this review is to understand first-language acquisition, especially the acquisition of syntax. At the core of any explanatory theory of language acquisition must be a theory of how, given the input available to a child, the child produces a grammar that is complex enough to account for the facts of productive adult language use. This is fundamentally a computational process, and computational analysis can provide some insight into how it is possible.

1.1. Empirical Research Versus Theoretical Research

An important contrast in computational methodologies is between empirical and theoretical approaches. In the empirical approach, a computer program is run on a specific corpus of data—typically of written text in a particular language—and the program is evaluated through a comparison of the output produced with some linguistically annotated text (a treebank). Often, the program does not produce a single grammar that is evaluated but rather directly produces some tree or dependency structures that are then compared with the treebank. In other cases, the program produces some grammar, typically a probabilistic grammar, that can then be used to produce analyses for the syntactically annotated sentences in the evaluation treebank. This type of research is best exemplified by the influential work of Klein & Manning (2004).

The alternative methodology is theoretical: One specifies a learning algorithm, and then proves that under certain conditions it can learn, in a precise technical sense, all grammars in some class of grammars. The algorithm may be implemented and tested, but this is not a necessary part of this approach.

Both approaches have some obvious advantages and disadvantages. First, the empirical approach relies on the existence of appropriate corpora; these corpora need to be large, and syntactically annotated at least in part for evaluation purposes. Such corpora will exist only for a small number of languages—for many interesting languages, especially those that are moribund or dead, we will never have appropriate corpora. Even for English, the available corpora of child-directed speech (MacWhinney 1995) are limited in a number of ways. Second, the syntactic annotations are not theoretically neutral. It is uncontroversial, at least among linguists, that sentences in natural language have some structure, but whether that structure is a constituent structure tree, a dependency structure or a derivation tree of a minimalist grammar, some typed feature structure, or a proof net or some other more exotic option is an open question. The lack of consensus about syntactic analyses was not widely understood until large-scale syntactic annotation projects began (Marcus et al. 1993). Third, linguists are limited by current computational infrastructure. Even the most powerful computers are much less powerful than the human brain; running large-scale experiments can involve computing costs that are not currently feasible. Finally, even when the algorithms do work, we do not know why they work or what properties of the languages they rely on.

To the extent that the corpora are naturally occurring samples of actually occurring language data, and to the extent that the learned grammar is adequate, we can be confident that the method works on a given amount of data. Even with a mathematical proof of the correctness of a learning algorithm, there may be hidden assumptions that call into question the validity of the algorithm. However, subject to these assumptions, the method of mathematical proof will give us the strongest possible guarantees. Moreover, we will often have a precise understanding of the properties of the grammars and languages that allow them to be learned, which gives us confidence that these algorithms can learn beyond the small sample of natural languages for which we have suitable corpora. For further discussion of the respective merits of these two approaches, see Niyogi (2006, pp. 37–38).

1.2. Survey of Theoretical Work

There is a fair amount of purely theoretical work on algorithms that learn context-free grammars (CFGs), much of which is based on inefficient enumerative algorithms. For example, Horning (1969) showed that probabilistic CFGs can be learned, and this line of research has been extended to essentially all possible grammars (see, e.g., Chater & Vitányi 2007). There is some research on learning categorial grammars under certain assumptions (Kanazawa 1994) but, again, using algorithms that are inefficient.

Efficient algorithms are few and far between: Beyond the methods I discuss in this review, I note especially the neglected paper by Shirakawa & Yokomori (1993), which prefigures some current approaches, and work by Yokomori (2003) on learning very simple grammars, another small class of context-free languages.

1.3. Survey of Empirical Work

This article is largely concerned with algorithms that have some theoretical guarantees of their correctness; these differ in a number of respects from other well-known approaches in the literature, which take an empirical perspective. This line of research has a very long history going back to Lamb (1961), which I do not attempt to survey exhaustively. Typically, these algorithms are based on heuristics of various types, and are evaluated by being run on corpora of child-directed speech, or other general corpora, and testing the resulting tree structures against sentences that have been manually annotated with some type of syntactic structure.

Perhaps the best-known approaches derive from Klein & Manning (2004), whose Dependency Model with Valence (DMV) has been widely influential. This model uses a dependency grammar, rather than a phrase-structure grammar like a CFG, together with a probabilistic model of how many arguments each head can take (the valence). Subsequent research has extended this model in various ways (Headden et al. 2009, Spitkovsky et al. 2010) and has been tested on quite a range of different languages. Another approach is Alignment Based Learning (van Zaanen 2000), which considers pairs of sentences and tries to align them to identify constituents; these constituents are then used to construct appropriate tree structures. For a detailed review of these and other influential approaches, see Heinz et al. (2015).

In the more linguistically oriented literature, there are several different families of approaches. One focuses on the acquisition of individual lexical items or pieces of linguistic information during the course of language acquisition; for example, Pearl & Sprouse (2012) look at the acquisition of syntactic islands, and Alishahi & Stevenson (2008) look at the acquisition of early argument structure. Another important line of research consists of parametric models of language acquisition, within the Principles and Parameters model of generative syntax. Here the learner needs to set a small finite number of binary parameters that specify certain properties of the target grammars (Gibson & Wexler 1994, Sakas & Fodor 2001, Yang 2002). Such models assume that much of the structure of the grammar is provided by the innate hypothesis space of the learner, rather than being learned from the input.

An alternative modeling assumption is that the learner has access to form/meaning pairs; this brings the model closer to something like semantic parsing (Kwiatkowski et al. 2010). Finally, some models use a limited amount of supervision. For example, Bisk & Hockenmaier (2013) use a model that has some information about lexical categories that can be used to initialize a categorial grammar.

1.4. Outline

The rest of this review is organized as follows. In Section 2 I consider algorithms that can learn classes of CFGs, starting with a very simple learning algorithm for a particular subclass, the substitutable context-free languages. I then consider a class of mildly context-sensitive grammars in Section 3, and then an even more powerful class of grammars that have an intrinsic copying operation in Section 4. In Section 5 I turn to the problem of learning the syntactic structure of a language, and in Section 6 I discuss the issue of indirect negative evidence and probabilistic learning. Finally, in Section 7 I briefly discuss the acquisition of semantics.

2. BASIC TECHNIQUES FOR CONTEXT-FREE GRAMMARS

A learning algorithm outputs grammars. So the starting point must be to consider the sort of grammar that is output.

I begin with the most basic model—the CFG. A CFG can model the most fundamental hierarchical aspect of language but is clearly inadequate in several respects for language (Chomsky 1956, Huybrechts 1984, Shieber 1985). Still, studying this limited class yields some insights, and because the learning approaches generalize from CFGs to richer formalisms (see Section 3), I explain the approaches using this well-understood formalism. It is helpful to use a general and nonlinguistic grammar formalism, like a CFG, which can be used to model other nonlinguistic data (such as biological sequence data), for practical reasons: In order to prove theorems about learning formalisms it is necessary, or at least helpful, that the formalisms be simple and mathematically tractable.

CFGs have a very strict locality condition: All of the information relevant to the well-formedness of a derivation is stored in a single object—a nonterminal symbol. These nonterminals can be thought of as crude syntactic categories. A CFG consists of a finite set of these nonterminals and some simple rules of the form $A \rightarrow \alpha$, where A is a nonterminal and α is a string of terminals and nonterminals. The terminal symbols can be thought of as words.[1] S refers to the start symbol of the grammar. I consider the derivation as a process that works either top down or bottom up,

[1] We assume that we can do word segmentation.

either where a nonterminal A is rewritten as the string α or where the elements of α are combined, or merged, to form an object of category A. This process can be repeated indefinitely; typically the set of possible derivations is infinite, and in general in these cases the language, the set of strings of words generated by the grammar, is also infinite. Below, I use $A \overset{*}{\Rightarrow}$ to signify that the nonterminal A can generate or derive the string of terminals w.

The term context-free means that what is derived from a nonterminal is independent of how that nonterminal is derived. In other words, given a derivation of the form $S \overset{*}{\Rightarrow} lAr$ and any derivation of the form $A \overset{*}{\Rightarrow} w$, one can combine the two to form a derivation $S \overset{*}{\Rightarrow} lAr \overset{*}{\Rightarrow} lwr$. In a learning situation, the derivations are not observed. We see only the final result of the derivation process; we do not observe the category A or the subderivation $A \overset{*}{\Rightarrow} w$; and we see only the final string lwr, without even the information that it is segmented or bracketed into parts in that way. In order to learn, we need to recover these syntactic categories, or some suitable proxy for them. Their primary syntactic role is to specify the compatibility relation between the two parts of a complete derivation: between a derivation context $S \overset{*}{\Rightarrow} lAr$ and a subderivation $A \overset{*}{\Rightarrow} w$. Therefore, it is natural to look at the relation between possible derivation contexts and possible subderivations as a source for these categories.

The derivation context in this sense is the rest of the derivation that takes some subtree and incorporates it into a whole sentence. The derivation contexts, in the current case involving only CFGs, correspond exactly to the set of possible pairs of strings (l, r), consisting of the context in which a nonterminal A can be derived. In other words, given a nonterminal A that derives a string w, the sentence of which it is part can only be of the form lwr for some strings l and r; the derivation process of a CFG cannot duplicate the string, reverse it, delete it, or insert strings in the middle. All the CFG can do is concatenate strings together. As discussed in Sections 3 and 4, more complex grammar formalisms in general will have a richer notion of context.

It is convenient to write these contexts as a sentence with a gap $l\Box r$. The possible subderivations correspond only to individual strings. From this perspective, a nonterminal such as A specifies that any derivation context of A can be combined with any subderivation of A; in other words, a set of contexts can be combined freely with a set of strings. The learning problem then involves finding sets of strings and sets of contexts such that any of the set of strings can be combined with any of the set of contexts to form a grammatical sentence. It has been proved (Clark 2013) that any grammar that does not have redundant nonterminals[2] will have nonterminals that define maximal string set/context set pairs. In simpler terms, we can restrict ourselves to looking for grammars in which the nonterminals correspond to sets of contexts and corresponding sets of strings, where neither of these sets can be increased without violating the condition that each context can combine freely with each string—the context-free property of the grammar. These maximal decompositions are called syntactic concepts, and this restriction reduces the seemingly intractable problem of grammar induction to the much more manageable problem of clustering strings and contexts. Given these concepts, it is easy to write down a grammar that will generate some subset of the language that is being learned, and given a sufficiently large set of these concepts—for example, one that includes the concepts corresponding to the nonterminals of the original grammar—the grammar will generate all and only the strings generated by the original grammar. This very high level description is not yet a specific algorithm, but a number of concrete algorithms have been developed that exploit these techniques to learn some class of CFGs under various conditions. I briefly summarize the various types of results.

[2] A nonterminal is redundant if it can be merged with another nonterminal without changing the language generated by the grammar.

2.1. Substitutable Languages

I begin with the result that initiated the current lines of research. Clark & Eyraud (2007) showed for the first time that results that had been well understood for the inference of regular grammars (Angluin 1982) could be extended to learning CFGs. This paper presents a learning algorithm for a subclass of CFGs. The algorithm relies on a simple property of the language called substitutability, which has a certain intuitive plausibility. Suppose the learner hears the following two sentences:

(1a) Look at the dog.
(1b) Look at the cat.

The learner reasons as follows: These two examples differ only in that *dog* and *cat* have been interchanged. Alternatively, one can say that both words appear in the same context, namely *look at the* □. From this fact the learner assumes that the two words are completely mutually substitutable: Wherever one sees the word *cat* in a grammatical sentence, one can replace it with the word *dog* and the result will be grammatical, and vice versa. The algorithm uses a very elementary form of analogical reasoning based on distributional similarity: *cat* and *dog* are distributionally similar in one respect, so they are distributionally similar in all respects.

Languages are substitutable if this inference is valid—if whenever two strings occur in the same context they are completely mutually substitutable. This idea has some prehistory (Myhill 1950, Chomsky 1959). Many formal languages such as propositional logic are substitutable, but natural languages are not. Consider the following examples:

(2a) Makoto is a mathematician.
(2b) Makoto is Japanese.
(2c) Makoto likes Japanese food.
(2d) * Makoto likes a mathematician food.

The first two examples show that *a mathematician* and *Japanese* occur in the same context, but the last two examples show that these two strings are not completely substitutable. Such examples are easy to find, so a learner using this principle will overgeneralize, and start to generate sentences such as example *2d*.

Note that mutual substitutability is an equivalence relation, classically called the syntactic congruence. Thus, one can form equivalence classes of strings, sets of strings that are completely substitutable or congruent, using as a guide those pairs of strings that occur in a context together. These clusters of strings formed from fragments of the observed utterances form the basis of the grammar. Where u refers to some individual strings, I use the notation $[u]$ to refer to the set of all strings that are completely substitutable for u.

For languages that are substitutable, the congruence classes correspond to the syntactic concepts of the language, so any reasonable grammar for the language must have nonterminals that correspond to these congruence classes. The learner therefore produces a nonterminal for each of the clusters that it forms, and trivially produces all binary rules according to the following schemas.

For all a that are words, we have the production

$$[a] \rightarrow a\,.$$

For all u and v that are strings, we have

$$[uv] \rightarrow [u][v].$$

These formulae are somewhat opaque. The notation $[u]$ represents the congruence class of u, the set of all strings that are completely mutually substitutable for u. The nonterminals I use correspond to these sets of strings. The nonterminal corresponding to $[u]$ should generate all of the strings in this set. The first schema then says simply that every congruence class that contains a single word has a nonterminal that can produce that word. This production is sound in the logical sense because of course a is in $[a]$. The second schema says that one can combine any element equivalent to u and any element equivalent to v and the result will be equivalent to the concatenation of u and v. This, again, is sound because of the mathematical fact that $[uv] \supseteq [u][v]$, for any strings u and v. Thus, the productions used in the grammar are justified by the mathematical properties of the sets of strings that the nonterminals represent. The soundness of these productions underlies the proofs of the correctness and effectiveness of the algorithm and the related, more powerful algorithms that I discuss in the next sections.

This learner produces a grammar that will, as the amount of data it sees increases, rapidly converge to a grammar that generates exactly the right language for all languages that are both substitutable and context free. It has several desirable characteristics.

- It is a very simple and theoretically well-motivated algorithm.
- It is computationally efficient: We can easily compute the hypothesis grammar from the input data.
- It is provably correct: It is mathematically guaranteed to converge to a correct grammar, a grammar that generates the right set of strings.
- Moreover, it is efficient in the amount of data used so that it is guaranteed to converge rapidly.
- It uses only positive data and makes no assumptions about how the data are generated.[3]
- The class of languages that can be learned has a simple characterization as the set of all context-free languages that are also substitutable. This class contains an infinite number of languages, infinitely many of which are infinite.

It is important to note that natural languages are clearly neither substitutable nor, in general, well described by CFGs; however, the simplicity of this algorithm makes it a good starting place, as the same ideas are used in the more sophisticated algorithms to which I now turn.

2.2. Learning with Membership Queries

The algorithm for learning substitutable CFGs uses only positive examples from the language, and will learn even when the examples are drawn adversarially.[4] Under this model the learner cannot control overgeneralization, because the absence of an example from the input is not reliable information—it may be an example that is being purposely delayed to confuse the learner.

However, there is no reason to think that the input to a child is carefully constructed to mislead the child, so this learning model is unrealistically hard, and as a result the classes of languages that can be learned are much too small (Clark & Lappin 2011). Therefore, it is appropriate to make the learning problem a bit easier—perhaps unrealistically easy—by allowing the learner to use what are called membership queries. In this learning model, the learner can pick a string and ask whether that string is in the target language or not. Of course, although the child is not completely passive, and can and does generate new sentences during the learning process, the child does not get direct feedback about the well-formedness of the utterances he/she produces, although he/she may receive some indirect indications. I discuss the validity of this assumption in Section 6 and show how these queries can be replaced with stochastic evidence, but here I assume

[3] Other than the most trivial assumption that the data come from the language we are trying to learn.

[4] Chosen by an infinitely powerful adversary who wants to make the learner fail.

that these queries are available, and note that it simplifies the analysis of the algorithms, at the cost of some plausibility. Importantly, using these queries, much larger classes of CFGs can be learned.

There are two families of algorithms that use what are called primal and dual techniques. These differ in how the syntactic categories are defined. For substitutable languages, one chooses congruence classes: sets of strings that are completely mutually substitutable. We are now considering languages that are not necessarily substitutable, so one must use as categories sets of strings that are mutually substitutable some of the time, but not necessarily all of the time. Two methods can be used to define these sets. The first, the dual method, uses a small, finite set of contexts. This set of two or three contexts, say, will define the set of strings that occur in all of these contexts. These strings will not necessarily be congruent. They will occur in some of the same contexts—at the very least, the set of contexts that are used to define the nonterminal.

The alternative method, the primal method, uses a small set of strings rather than contexts to define the nonterminal. I use W to refer to this small finite set, and the set of strings generated by the nonterminals should correspond to the set of all strings that can occur in all of the contexts that all of the elements of W can occur in.

It is helpful to consider an example where this representational power is useful. Consider simple cases of lexical ambiguity in English. For example, *rose* can be both a count noun and the preterite of the verb *rise* (among other possibilities, such as a proper name or a color, which are not discussed here). The learner is not directly informed that *rose* is syntactically ambiguous. Instead, she is exposed to various examples such as the following:

(3a) She rose from the table.

(3b) That is a lovely rose.

The congruence class containing *rose* contains only strings that can occur in all of the contexts that *rose* can occur in, so the only words that it contains are those that are also ambiguous between count nouns and preterite verbs; *rose* is not congruent to *lily*, nor is it congruent to *floated*. As a syntactic category, therefore, the congruence class of an ambiguous word is very undesirable. The categories that we do want, such as that of a singular count noun, must therefore contain strings that are not perfectly distributionally identical. In this case, one can construct a suitable category either primally or dually. The primal approach involves taking two words, say, *rose* and *lily*, and defining the category as corresponding to all words that can occur in all contexts in which both *rose* and *lily* can occur. For example, the word *floated* is not part of this collection, but the word *carnation* is. Whether other count nouns such as *book* are in the class depends on how precisely we define the threshold for grammaticality and whether it includes semantic well-formedness constraints.

The dual approach involves choosing a small set of contexts that serve to select the elements of the category. One could choose the contexts *that is a lovely* □ and *that* □ *is lovely*. Neither one on its own will select the right set of strings, as the first context allows various extra strings—for example, *rose but I prefer orchids*. Similarly, the string *rose is horrible but the orchid* can occur in the context *that* □ *is lovely*. Crucially, by taking more than one context, one can eliminate the excess strings by effectively intersecting the sets of strings that can occur in each individual context.

Using these methods, one can define simple learning algorithms. The classes that can be learned correspond to grammars whose nonterminals can be defined either primally or dually. If a grammar has the property that each nonterminal can be defined primally with *k* strings, where *k* is some small number such as two or three, then the grammar has the *k* Finite Kernel Property

(FKP). Similarly, if every nonterminal can be defined with k contexts, then the grammar has the k Finite Context Property (FCP). There are algorithms that can efficiently learn all grammars with the k FKP or the k FCP for any fixed value of k (Clark & Yoshinaka 2013).

These algorithms can learn many but not all context-free languages. In particular, all regular languages are included in the smallest of these classes, when k is one, and most standard examples of CFGs are also in this same class, such as the language $\{a^n b^n \mid n > 0\}$, which consists of any number of as followed by an equal number of bs. Importantly, these algorithms cannot learn all context-free languages: The language $\{a^m b^n \mid m \neq n\}$, which contains a number of as followed by a different number of bs, is not in any of the learnable classes for any value of k. However, this sort of language does not correspond to any phenomenon found in natural languages. Apart from these types of examples, essentially all grammars seem to lie in these learnable classes for small numbers of k.

Although this approach is interesting, it is very far from an adequate solution to the problem discussed here. This research has several important limitations. First, the class of representations, CFGs, is inadequate for natural languages. Second, the use of membership queries is highly questionable. Finally, this approach concerns only weak learning—the problem of learning a set of strings, rather than a set of structures—and as such has been criticized as being of no linguistic interest (Berwick et al. 2011). The question is whether these issues are intrinsic limitations of the distributional approach, as Berwick et al. (2011) argue, or whether they can be overcome using appropriate modifications and extensions. Fortunately, we now know that these problems can be solved. In the following sections, I discuss how these problems have been overcome in subsequent research.

3. MILDLY CONTEXT-SENSITIVE GRAMMARS

The inadequacy of CFGs (Huybrechts 1984, Shieber 1985) prompted intense research on slightly more powerful formalisms that were efficient computationally yet powerful enough to describe natural languages. Joshi et al. (1990) introduced the notion of a mildly context-sensitive class of grammars to try to define the desirable properties of a class of grammars. The class of multiple context-free grammars (MCFGs) (Seki et al. 1991) is, in my view, the most natural extension of CFGs, and it is weakly and strongly equivalent (Michaelis 2001) to a class of minimalist grammars (Stabler 1997) that is widely accepted as a formalization of contemporary syntactic models. MCFGs have nonterminals that can derive not only strings but also pairs of strings, or indeed tuples of higher arity. The productions in the grammar can then interleave components of the strings, thus naturally modeling the displacement or movement operations that are common in syntax, where a constituent appears in a surface position that is displaced from where it is originally introduced.

In a series of papers, Yoshinaka (2010, 2011) showed that the methods of learning CFGs can be extended naturally to MCFGs. In order to extend the learning ideas to the MCFG case, one must consider the derivation contexts of an MCFG. A nonterminal might derive a pair of strings $\langle u, v \rangle$; the derivations of the MCFG[5] will then combine these into a final string of the form $lumvr$. The derivation context is thus a sentence with two gaps, $l\Box m\Box r$, where the first component of the derived tuple will be inserted in the first gap and the second component in the second gap. A linguistic example would be a sentence with simple *wh*-movement, such as example 4*a*, which could be analyzed as having a nonterminal deriving the pair of strings in example 4*b*, the derivation context of which would be example 4*c*. In this case, the first component represents a

[5] I consider here only a restricted normal form of MCFGs.

moving constituent, and the second component represents the partially constructed constituent out of which the first component is moving.

(4a) Which book did John read?

(4b) ⟨Which book, read⟩

(4c) □ did John □

The learning algorithms can exploit the relationship between the derivation contexts, which in this case are sentences with two gaps, and the subderivations, which are taken to be pairs of strings. In this way, the algorithms for CFGs can be directly translated into algorithms for learning these mildly context-sensitive grammars.

4. COPYING

It seems necessary to have some copying as a primitive operation in the grammar (Kobele 2006). Parallel MCFGs (Seki et al. 1991) are an extension of MCFGs that allow the bottom-up derivation process to copy a constituent so that it will appear two or more times on the surface. This process allows a natural treatment of various phenomena, such as contrastive focus reduplication in English (Ghomeshi et al. 2004), various types of *Suffixaufnahme*, reduplication in morphology, and relative clause duplication in Yoruba and Wolof, the treatment of which, without such an operation, would be difficult or impossible. Such an additional operation means that the grammars can represent languages that are not semilinear, which means that they are more powerful than the mildly context-sensitive languages.

So, for example, in English one can say "I want some salad salad." In this case, the reduplication of *salad* gives it the meaning of a prototypical salad consisting of lettuce, say, as opposed to fruit salad. In order to model this without redundancy, one needs an operation that can duplicate or copy. Again, these richer classes of grammars can be learned (Clark & Yoshinaka 2013) by using an appropriately richer model of derivation and defining the appropriate types of derivation contexts. With the addition of copying, the derivation process may copy parts of the string, so the derivation context is in general a function that may copy some of the arguments.

For example, if we have a subderivation that generates a string w, the rest of the derivation may copy that string and insert other material, so that the whole sentence is $lwmwr$, where there are two occurrences of w. In this case, the context is a function that maps w to $lwmwr$, which we can write as $f(x) = lxmxr$. In the case of the example above, the context of the word *salad* is the function that maps w to "I want some w w." From this perspective, the simple contexts of a CFG that we write as $l\Box r$ are nothing but trivial functions of the form $f(x) = lxr$.

5. LEARNING SYNTACTIC STRUCTURE

So far I have discussed only algorithms that, from a finite amount of information about the strings in a formal language, can produce a grammar that correctly generalizes to the whole formal language. However, a language is not only a set of strings but also a relation between strings and acceptable interpretations for these strings. Because the learners discussed above do not have any information about the available interpretations, it might seem that we cannot even discuss this issue and that we need to consider learners that have access to semantic information (I address this topic in Section 7, below). It is possible, however, to consider learners that are required to learn a grammar that generates not only the right set of strings but also the right set of structural descriptions. In the case of learning CFGs, this means that, given a grammar that we are trying to learn, G_*, we want the learner to produce a hypothesis, $\hat{G}$, that is not only weakly equivalent to the G_* but also strongly equivalent. In the case of CFGs, we want a grammar that is isomorphic to $\hat{G}$, identical except for the symbols used for the nonterminals. Of course, such a grammar will clearly also be weakly equivalent to the target grammar. Such a learner is called a strong learner, by analogy with the traditional notion of strong generative capacity (Berwick 1984, Miller 1999). For a learner to be a strong learner, it must have a clearly defined notion of syntactic structure that can be derived from the formal language itself, as a set of strings, and not from some arbitrary grammar. Because the learner has information only about the language, and not the grammar, this approach requires the existence of a canonical grammar for each formal language: a unique grammar for each set of strings.

To understand what this means, consider the following thought experiment. Take an existing natural language, English. Is it possible for there to be another language, say, Schmenglish, that has exactly the same set of grammatical sentences but different syntactic structures and different degrees of ambiguity? If so, every sentence in English is grammatical if and only if it is grammatical in Schmenglish, but some sentences in English are, for example, unambiguous whereas they are ambiguous in Schmenglish. Intuitions may vary about this, but this model assumes that such pairs of languages are impossible. In other words, the class of natural languages cannot contain two grammars that are weakly equivalent but not strongly equivalent. This is a necessary condition for a class of languages to be strongly learnable only from the strings. This thought experiment reveals some fundamental limits on the possibility of strong learning, which may ultimately be too demanding a learning model.

Current proposals (Clark 2014, 2015) show that such canonical grammars can be based on the discovery of primes, irreducible elements of structures associated with the languages. In many simple cases, these canonical grammars correspond to minor modifications of the natural grammars that one would write down for a given set of strings. On this view, then, syntactic structure corresponds to a deep mathematical property of the distributional patterns in the sets of strings. The grammars are not arbitrary but rather are determined by the sets of strings that they generate.

The structures in this case correspond to the derivation trees of the relevant grammar formalisms: CFGs, MCFGs, or other formalisms. Although linguists tend to prefer the derived trees as being more salient and natural as representational structures, current thinking in the mathematical linguistics community is that the derivation trees are more fundamental and are sufficient to support semantic interpretation, at least in the case of minimalist grammars and tree-adjoining grammars (Graf 2013; Kobele 2011, 2015). Although still preliminary, this research provides an intriguing possible account of the acquisition of syntactic structure.

6. DIRECT AND INDIRECT NEGATIVE EVIDENCE

An enduring debate in language acquisition involves the role of negative evidence (Marcus 1993) and how children can recover from overgeneralization errors—that is, how children determine that their grammar generates too much, if they are exposed only to grammatical sentences and have received no direct feedback about the ungrammatical sentences. This debate has given rise to an extensive literature under various names: the logical problem of language acquisition, the subset problem, and so on (Fodor & Sakas 2005).

For both historical and technical reasons, the theory of learning grammars developed along rather different lines (Osherson et al. 1986) than that of mainstream machine learning. As a result, the former theory has generally used nonprobabilistic learning models, such as Gold's influential identification in the limit model (Gold 1967). In this model, it is indeed hard for a learner to recover from an overgeneralization error, as the absence of a sentence from the data seen so far provides no information at all about whether it is ungrammatical. This is because the data in the Gold model may be generated adversarially and the adversary can delay any sentence indefinitely. In more realistic learning models, the data are generated randomly, and under these circumstances the absence of data from the input is helpful (Horning 1969, Clark & Lappin 2009). This is sometimes called learning from indirect negative evidence.

Regarding learning in general, recovering from overgeneralization does not pose a problem. In probabilistic models of learning, the learning algorithms typically try to assign high probabilities to those events that actually occur. A model that assigns a lot of probability mass to events that do not occur will typically have a lower rating than one that does not. Indeed, the general problem in machine learning is the opposite one: the problem of overfitting the data that have been observed, and failing to generalize. For further discussion, see Regier & Gahl (2004), Chater & Manning (2006), Perfors et al. (2011), and Chater et al. (2015).

The more advanced methods discussed above rely on the use of membership queries; these allow the learners to obtain direct evidence about which strings are not in the language. This simplifies the algorithm and allows for elementary proofs without the use of complex probabilistic inequalities. However, for some researchers this may be too idealized. One of the reasons it seems to be a reasonable assumption is that the use of membership queries can be replaced with a reliance on stochastic data. For example, in the classical theory of learning regular languages, the original algorithms that used queries (Angluin 1987) can be modified directly to learn from positive data (Clark & Thollard 2004), and other efficient stochastic learning models have now been developed (Denis et al. 2004, Mossel & Roch 2006) that can learn the whole class of regular languages from stochastic data using rather different techniques.

Such techniques have been extended to CFGs (Clark 2006). This initial result is very limited, but Shibata & Yoshinaka (2013) extended it to classes of CFGs that are large enough to generate all regular languages, to the class of 1-FKP languages, and to a subclass of 1-FCP languages. This research indicates that in general the membership query–based model is not absurd. In order to learn in the query-based model, one needs to be able to efficiently construct a hypothesis grammar from the data; the other problem, of controlling overgeneralization, can be handled using standard techniques (see, for example, the very powerful Bayesian techniques explored by Chater & Vitányi 2007).

However, a note of caution is in order. When we assume that the data are generated randomly, a question arises as to the nature of the process of generation, and the particular probabilities that each sentence has. The particular probability distribution that is used does have a crucial effect on learnability, and it is not the case that for every possible distribution the learner will succeed. For many learners, one can construct distributions of examples such that the learner is likely to fail. Positive results assume that the distributions are causally based on the grammar being learned—which is, of course, reasonable, as the sentences from which the learner learns are generated by speakers of the language—and rely on other parameters that measure in some sense how easy it is to learn from those examples. For example, if there are two nonterminals in a grammar that are very close together distributionally, it will be hard for a learner to tell them apart. Such problems will be harder to learn and may require many more examples for the learner to succeed. Only when the distributions are reasonably helpful is the learner able to learn rapidly—from a small amount of data.

7. THE ROLE OF SEMANTICS

There are many models of learning in which the input to the learner consists of sound/meaning pairs. In the later phases of acquisition, children can under some circumstances infer the meaning of an utterance from their already-acquired syntactic knowledge and their knowledge of the communicative intentions and the situational context, even when the utterance contains novel words or constructions. This justifies a learning model where the learner has access to sound/meaning pairs (Crain & Pietroski 2001). In this situation, the learning problem becomes significantly harder. Although the information source is richer, because we have sound/meaning pairs rather than only sounds, the range of operations that are used in semantic derivations are wider and more powerful than those used in syntax alone. In this context, the use of copying becomes crucial; for example, coordination often involves some sort of copying as traditionally modeled.

For instance, example 5*a* has a meaning that can be represented formally as something like example 5*b*, and the VP has a meaning something like the lambda expression in example 5*c*, which copies the meaning of the subject NP, X. Clearly, the constant **olga** appears twice in example 5*b* but only once in example 5*a*. At some point it is copied.

(5a) Olga likes sushi and sashimi.

(5b) $\mathbf{like}(\mathbf{olga}, \mathbf{sushi}) \wedge \mathbf{like}(\mathbf{olga}, \mathbf{sashimi})$

(5c) $\lambda X.(\mathbf{like}(X, \mathbf{sushi}) \wedge \mathbf{like}(X, \mathbf{sashimi}))$

A model that can learn these sorts of sound/meaning mappings is necessary in order to have a satisfactory acquisition model. Unrestrained copying, though, leads to computational intractability, so in order to obtain an efficient learning algorithm we need to consider some constraints on the degree of copying that the grammar uses, while still allowing the richness that is necessary for natural language semantics.

Distributional learning methods can be extended to at least a partial solution to the problem (Yoshinaka & Kanazawa 2011) via the use of abstract categorial grammars, a powerful grammatical formalism that allows a unified representation of both sound and meaning (de Groote 2001). These grammars have an abstract syntax tree and separate components that construct the sounds and the meanings from the abstract tree. One can construct a learning algorithm that learns from the resulting pairs. However, the class is still much too limited to be appropriate for natural language syntax; extending the grammars that can be learned to allow copying on the semantic side is an active area of research (Clark et al. 2016).

8. DISCUSSION

The task of learning grammars from strings is not impossibly hard. There are some conceptually simple algorithms that are capable of carrying out these tasks. Even the simplest models show that there is no reason to think of parameter-setting models as being logically necessary for the acquisition of syntax, as some researchers have claimed (Boeckx 2006, Yang 2008).

Chomsky (1986, p. 52) famously said:

> To achieve descriptive adequacy it often seems necessary to enrich the system of available devices, whereas to solve our case of Plato's problem we must restrict the system of available devices so that only a few languages or just one are determined by the given data. It is the tension between these two tasks that makes the field an interesting one, in my view.

Here I attempt to resolve this tension by choosing a restricted system of grammars defined in two parts. The first part involves choosing a sufficiently large class of mildly context-sensitive grammars, or some moderate extension of them to include copying, and the second part involves restricting them to those that satisfy some distributional regularity that allows learnability—some variant of the FKP or FCP. The combination yields a class of grammars that is both descriptively adequate and learnable, and can thus attain explanatory adequacy.

The distributional regularities that we rely on are definitely oversimplified. It is not plausible that the child learner literally represents each syntactic category via a small set of contexts or strings. Rather, I conjecture that the child maintains some statistical ensemble over the contexts and strings of the nonterminal using local distributional features. The specific proposals discussed above should be considered analytically tractable idealizations of the true situation. The model relies on particular string properties of the grammars, in particular on the properties of the phonological form of the words. This is not something that classical generative grammar has paid much attention to, but it is clearly crucial for learnability. If every word in a language was pronounced the same, then the language would clearly be unlearnable and indeed incomprehensible.

Both the innate endowment of the child and the input data are essential: One is not more important than the other. The proposals discussed in this review do not reject the innate endowment; on the contrary, they are an explicit theory of Universal Grammar and the Language Acquisition Device. Although these methods share some technical similarities with the oversimplistic distributional methods of the American structuralists (Wells 1947, Harris 1951), they are very different in purpose, method, and degree of mathematical completeness. The structuralists were concerned primarily with discovery procedures, whereas the research discussed here involves models of language acquisition.

Modern research requires more than purely qualitative models: Explicit computational models of language acquisition are crucial for allowing the integration of linguistic theory with developmental data, neural data, and data drawn from corpora of child-directed speech. This has not been possible until now, because of the absence of a good theoretical understanding of the acquisition problem. The algorithms presented here show that there are efficient, well-understood learning algorithms with good theoretical guarantees. Although they are still very abstract, I hope that they can form a starting point for more plausible models of language acquisition that connect with the developmental evidence.

SUMMARY POINTS

1. There are efficient algorithms for learning large classes of grammars from strings.
2. These grammars can be CFGs or mildly context-sensitive grammars, or even grammars that include copying as a primitive operation.
3. The classes of grammars include the class of natural languages, even under the most pessimistic assumptions about their generative capacity.
4. The learning algorithms are efficient in an idealized mathematical sense. This does not mean that they can learn as rapidly as human infants can.
5. Under some circumstances the syntactic structure can be learned without using semantic information.

FUTURE ISSUES

1. Learning the syntactic structure of grammars with movement is very challenging: It may be necessary to incorporate semantic information.

2. The theoretical analysis does not yet include a notion of syntactic feature, so the grammars that are learned are too large and redundant to be plausible.
3. More research is needed to understand how the acquisition of semantics can occur in parallel with the acquisition of syntax.

DISCLOSURE STATEMENT

The author is not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

I thank Shalom Lappin, Ryo Yoshinaka, Greg Kobele, and Makoto Kanazawa for many helpful discussions.

LITERATURE CITED

Alishahi A, Stevenson S. 2008. A computational model of early argument structure acquisition. *Cogn. Sci.* 32:789–834

Angluin D. 1982. Inference of reversible languages. *J. ACM* 29:741–65

Angluin D. 1987. Learning regular sets from queries and counterexamples. *Inf. Comput.* 75:87–106

Berwick R. 1984. Strong generative capacity, weak generative capacity, and modern linguistic theories. *Comput. Linguist.* 10:189–202

Berwick R, Pietroski P, Yankama B, Chomsky N. 2011. Poverty of the stimulus revisited. *Cogn. Sci.* 35:1207–42

Bisk Y, Hockenmaier J. 2013. An HDP model for inducing combinatory categorial grammars. *Trans. Assoc. Comput. Linguist.* 1:75–88

Boeckx C. 2006. *Linguistic Minimalism*. Oxford, UK: Oxford Univ. Press

Chater N, Clark A, Goldsmith JA, Perfors A. 2015. *Empiricism and Language Learnability*. Oxford, UK: Oxford Univ. Press

Chater N, Manning C. 2006. Probabilistic models of language processing and acquisition. *Trends Cogn. Sci.* 10:335–44

Chater N, Vitányi P. 2007. 'Ideal learning' of natural language: positive results about learning from positive evidence. *J. Math. Psychol.* 51:135–63

Chomsky N. 1956. Three models for the description of language. *IEEE Trans. Inf. Theory* 2:113–24

Chomsky N. 1959. Review of Joshua Greenberg's *Essays in Linguistics*. *Word* 15:202–18

Chomsky N. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. New York: Praeger

Clark A. 2006. PAC-learning unambiguous NTS languages. In *Grammatical Inference: Algorithms and Applications*, ed. Y Sakakibara, S Kobayashi, K Sato, T Nishino, E Tomita, pp. 59–71. Berlin: Springer

Clark A. 2013. The syntactic concept lattice: another algebraic theory of the context-free languages? *J. Logic Comput.* doi:10.1093/logcom/ext037

Clark A. 2014. Learning trees from strings: a strong learning algorithm for some context-free grammars. *J. Mach. Learn. Res.* 14:3537–59

Clark A. 2015. Canonical context-free grammars and strong learning: two approaches. In *Proceedings of the 14th Meeting on the Mathematics of Language* (*MOL 2015*), ed. M Kuhlmann, M Kanazawa, GM Kobele, pp. 99–111. Stroudsburg, PA: Assoc. Comput. Linguist.

Clark A, Eyraud R. 2007. Polynomial identification in the limit of substitutable context-free languages. *J. Mach. Learn. Res.* 8:1725–45

Clark A, Kanazawa M, Kobele GM, Yoshinaka R. 2016. Distributional learning of some nonlinear tree grammars. *Fundam. Inf.* 146:1–39

Clark A, Lappin S. 2009. Another look at indirect negative evidence. In *Proceedings of the EACL Workshop on Cognitive Aspects of Computational Language Acquisition*, pp. 26–33. Stroudsburg, PA: Assoc. Comput. Linguist.

Clark A, Lappin S. 2011. *Linguistic Nativism and the Poverty of the Stimulus*. Malden, MA: Wiley-Blackwell

Clark A, Thollard F. 2004. PAC-learnability of probabilistic deterministic finite state automata. *J. Mach. Learn. Res.* 5:473–97

Clark A, Yoshinaka R. 2013. Distributional learning of parallel multiple context-free grammars. *Mach. Learn.* 96:5–31

Crain S, Pietroski P. 2001. Nature, nurture and universal grammar. *Linguist. Philos.* 24:139–86

de Groote P. 2001. Towards abstract categorial grammars. In *Proceedings of the 39th Conference of the Association for Computational Linguistics*, pp. 252–59. Stroudsburg, PA: Assoc. Comput. Linguist.

Denis F, Lemay A, Terlutte A. 2004. Learning regular languages using RFSAs. *Theor. Comput. Sci.* 313:267–94

Fodor J, Sakas W. 2005. The subset principle in syntax: costs of compliance. *J. Linguist.* 41:513–70

Ghomeshi J, Jackendo R, Rosen N, Russell K. 2004. Contrastive focus reduplication in English (the salad-salad paper). *Nat. Lang. Linguist. Theory* 22:307–357

Gibson E, Wexler K. 1994. Triggers. *Linguist. Inq.* 25:407–54

Gold EM. 1967. Language identification in the limit. *Inf. Control* 10:447–74

Graf T. 2013. *Local and transderivational constraints in syntax and semantics*. PhD thesis, Dep. Linguist., Univ. Calif., Los Angeles

Harris Z. 1951. *Methods in Structural Linguistics*. Chicago: Univ. Chicago Press

Headden WP III, Johnson M, McClosky D. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 101–9. Stroudsburg, PA: Assoc. Comput. Linguist.

Heinz J, de la Higuera C, van Zaanen M. 2015. *Synthesis Lectures on Human Language Technologies: Grammatical Inference for Computational Linguistics.* San Rafael, CA: Morgan & Claypool

Horning JJ. 1969. *A study of grammatical inference*. PhD thesis, Comput. Sci. Dep., Stanford Univ., Stanford, CA

Huybrechts RAC. 1984. The weak inadequacy of context-free phrase structure grammars. In *Van Periferie naar Kern*, ed. G de Haan, M Trommelen, W Zonneveld, pp. 81–99. Dordrecht, Neth.: Foris

Joshi A, Vijay-Shanker K, Weir D. 1990. *The convergence of mildly context-sensitive grammar formalisms*. Tech. rep. MS-CIS-90-01, Dep. Comput. Inf. Sci., Univ. Pa., Philadelphia

Kanazawa M. 1994. *Learnable classes of categorial grammars*. PhD thesis, Comput. Sci. Dep., Stanford Univ., Stanford, CA

Klein D, Manning C. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of the 42nd Annual Conference of the Association for Computational Linguistics*, art. 478. Stroudsburg, PA: Assoc. Comput. Linguist.

Kobele GM. 2006. *Generating copies: an investigation into structural identity in language and grammar*. PhD thesis, Dep. Linguist., Univ. Calif., Los Angeles

Kobele GM. 2011. Minimalist tree languages are closed under intersection with recognizable tree languages. In *Lecture Notes in Computer Science*, vol. 6736: *Logical Aspects of Computational Linguistics*, ed. S Pogodalla, JP Prost, pp. 129–44. Berlin: Springer

Kobele GM. 2015. LF-copying without LF. *Lingua* 166:B236–59

Kwiatkowski T, Zettlemoyer L, Goldwater S, Steedman M. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1223–33. Stroudsburg, PA: Assoc. Comput. Linguist.

Lamb SM. 1961. On the mechanisation of syntactic analysis. In *Proceedings of the 1961 Conference on Machine Translation of Languages and Applied Language Analysis*, 2:674–85. London: Her Majesty's Station. Off.

MacWhinney B. 1995. *The CHILDES Project: Tools for Analyzing Talk*. Hillsdale, NJ: Erlbaum

Marcus GF. 1993. Negative evidence in language acquisition. *Cognition* 46:53–85

Marcus MP, Santorini B, Marcinkiewicz MA. 1993. Building a large annotated corpus of English: the Penn Treebank. *Comput. Linguist.* 19:312–30

Michaelis J. 2001. Transforming linear context-free rewriting systems into minimalist grammars. In *Logical Aspects of Computational Linguistics*, ed. P de Groote, G Morrill, C Retoré, pp. 228–44. Berlin: Springer

Miller P. 1999. *Strong Generative Capacity: The Semantics of Linguistic Formalism*. Stanford, CA: Cent. Study Lang. Inf.

Mossel E, Roch S. 2006. Learning nonsingular phylogenies and hidden Markov models. *Ann. Appl. Probab.* 16:583–614

Myhill J. 1950. Review of *On Syntactical Categories* by Yehoshua Bar-Hillel. *J. Symb. Logic* 15:220

Niyogi P. 2006. *The Computational Nature of Language Learning and Evolution*. Cambridge, MA: MIT Press

Osherson D, Stob M, Weinstein S. 1986. *Systems That Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists*. Cambridge, MA: MIT Press. 1st ed.

Pearl L, Sprouse J. 2012. Computational models of acquisition for islands. In *Experimental Syntax and Island Effects*, ed. J Sprouse, N Hornstein, pp. 109–31. Cambridge, UK: Cambridge Univ. Press

Perfors A, Tenenbaum JB, Regier T. 2011. The learnability of abstract syntactic principles. *Cognition* 118:306–38

Regier T, Gahl S. 2004. Learning the unlearnable: the role of missing evidence. *Cognition* 93:147–55

Sakas W, Fodor J. 2001. The structural triggers learner. In *Language Acquisition and Learnability*, ed. S Bertolo, pp. 172–233. Cambridge, UK: Cambridge Univ. Press

Seki H, Matsumura T, Fujii M, Kasami T. 1991. On multiple context-free grammars. *Theor. Comput. Sci.* 88:191–229

Shibata C, Yoshinaka R. 2013. PAC learning of some subclasses of context-free grammars with basic distributional properties from positive data. In *Lecture Notes in Computer Science*, vol. 8139: *Algorithmic Learning Theory*, ed. S Jain, R Munos, F Stephan, T Zeugmann, pp. 143–57. Berlin: Springer

Shieber S. 1985. Evidence against the context-freeness of natural language. *Linguist. Philos.* 8:333–43

Shirakawa H, Yokomori T. 1993. Polynomial-time MAT learning of C-deterministic context-free grammars. *Trans. Inf. Process. Soc. Jpn.* 34:380–90

Spitkovsky VI, Alshawi H, Jurafsky D, Manning CD. 2010. Viterbi training improves unsupervised dependency parsing. In *Proceedings of the 14th Conference on Computational Natural Language Learning* (*CoNLL2010*), pp. 9–17. Stroudsburg, PA: Assoc. Comput. Linguist.

Stabler E. 1997. Derivational minimalism. In *Lecture Notes in Computer Science*, vol. 1328: *Logical Aspects of Computational Linguistics*, ed. C Retoré, pp. 68–95. Berlin: Springer

van Zaanen M. 2000. ABL: alignment-based learning. In *Proceedings of the 18th International Conference on Computational Linguistics* (*COLING 2000*), pp. 961–67. Stroudsburg, PA: Assoc. Comput. Linguist.

Wells RS. 1947. Immediate constituents. *Language* 23:81–117

Yang C. 2002. *Knowledge and Learning in Natural Language*. New York: Oxford Univ. Press

Yang C. 2008. The great number crunch. *J. Linguist.* 44:205–28

Yokomori T. 2003. Polynomial-time identification of very simple grammars from positive data. *Theor. Comput. Sci.* 298:179–206

Yoshinaka R. 2010. Polynomial-time identification of multiple context-free languages from positive data and membership queries. In *Proceedings of the 10th International Colloquium on Grammatical Inference*, pp. 230–44. Berlin: Springer

Yoshinaka R. 2011. Efficient learning of multiple context-free languages with multidimensional substitutability from positive data. *Theor. Comput. Sci.* 412:1821–31

Yoshinaka R, Kanazawa M. 2011. Distributional learning of abstract categorial grammars. In *Lecture Notes in Computer Science*, vol. 6736: *Logical Aspects of Computational Linguistics*, ed. S Pogodalla, JP Prost, pp. 251–66. Berlin: Springer

Annual Review of
Linguistics

Volume 3, 2017

Contents

Errata

An online log of corrections to *Annual Review of Linguistics* articles may be found at http://www.annualreviews.org/errata/linguistics