Learnable Representations of Language
A course at ESSLLI 2010

Alexander Clark
Department of Computer Science,
Royal Holloway, University of London
alexc@cs.rhul.ac.uk

August, 2010

This is the course material for the course "Learnable Representations of Languages", at ESSLLI 2010.

This course will cover the basic theory of grammatical inference for richly structured languages; a course outline, subject to revision is:

Lecture 1 Fundamental methodological issues.

Lecture 2 Inference of regular languages and regular transductions

Lecture 3 Congruence based approaches to context free inference

Lecture 4 Dual approaches to context free inference and distributional lattice grammars.

Lecture 5 Multiple context free grammars and general algorithms for grammar induction.

The web page for this course is at this address:

`http://www.cs.rhul.ac.uk/home/alexc/esslli2010/index.html`

Course notes and slides will be made available there.

This document includes two more general papers, that give an overview of the material we will cover in the course.

Contents

1 Introductory material

The first lecture will cover the fundamental methodological issues. Key points are:

- Why is learnability important?
- The linguistic debates around learnability and the Argument from the Poverty of the Stimulus.
- Various ways of formalising learnability: in particular Gold style learnability in the limit and probabilistic learnability.
- Important negative results in learnability

The following paper is a draft chapter for the Handbook of the Philosophy of Linguistics, edited by Ruth Kempson, and is joint work with Shalom Lappin.

Computational Learning Theory and Language Acquisition

Alexander Clark[1] and Shalom Lappin[2]

[1] Department of Computer Science,
Royal Holloway, University of London
alexc@cs.rhul.ac.uk
[2] Department of Philosophy,
King's College, London
shalom.lappin@kcl.ac.uk

Acknowledgement. We are grateful to Richard Sproat for his careful reading of an earlier draft of this chapter, and for his invaluable suggestions for correction. Of course we bear sole responsibility for the content of the chapter.

1 Introduction

Computational learning theory explores the limits of learnability. Studying language acquisition from this perspective involves identifying classes of languages that are learnable from the available data, within the limits of time and computational resources available to the learner. Different models of learning can yield radically different learnability results, where these depend on the assumptions of the model about the nature of the learning process, and the data, time, and resources that learners have access to. To the extent that such assumptions accurately reflect human language learning, a model that invokes them can offer important insights into the formal properties of natural languages, and the way in which their representations might be efficiently acquired.

In this chapter we consider several computational learning models that have been applied to the language learning task. Some of these have yielded results that suggest that the class of natural languages cannot be efficiently learned from the primary linguistic data (PLD) available to children, through

A draft chapter for the Elsevier *Handbook of Philosophy of Linguistics*, edited by Ruth Kempson, Nicholas Asher, and Tim Fernando. This draft formatted on 7th July 2010.

domain general methods of induction. Several linguists have used these results to motivate the claim that language acquisition requires a strong set of language specific learning biases, encoded in a biologically evolved language faculty that specifies the set of possible languages through a Universal Grammar.[1]

In fact, when the assumptions underlying these models are carefully examined, we find that they involve highly implausible claims about the nature of human language learning, and the representation of the class of natural languages. Replacing these models with ones that correspond to a more realistic view of the human learning process greatly enhances the prospect for efficient language learning with domain general induction procedures, informed by comparatively weak language specific biases. Specifically, various procedures based on the ideas of *distributional learning* show that significant classes of languages can be learned.

2 Linguistic Nativism and Formal Models of Learning

The view that a set of strong language specific learning biases is a necessary condition for language acquisition can be described as *linguistic nativism*. This view has been endorsed by, *inter alia*, Chomsky (1965, 1975, 1981, 1995, 2000, 2005), Crain & Pietroski (2002), Fodor & Crowther (2002), Niyogi & Berwick (1996), Nowak *et al.* (2001), Pinker (1984), Pinker & Jackendoff (2005), and Yang (2002). It has been dominant in linguistics and cognitive psychology for the past fifty years. One of the central motivations for this view is the claim that if children were equipped only with domain general learning procedures of the sort that they employ to achieve many kinds of non-linguistic knowledge, they would not be able to acquire the complex grammars that represent the linguistic competence of native speakers. The argument takes domain general inductive learning of grammar to be ruled out by limitations on the primary linguistic data (PLD) to which children are exposed, and restrictions on the resources of time and computation available to them. This view is commonly known as the *argument from the poverty of the stimulus* (APS).

There are several different versions of the APS, each of which focuses on a distinct aspect of the way in which the PLD underdetermines the linguistic knowledge that a mature native speaker of a language acquires.[2] In this chapter we are concerned with the APS as a problem in formal learning theory,

[1] For a discussion of the relevance of current work in computational learning theory to grammar induction, see Clark & Lappin (2010a). For a detailed discussion of the connection between computational learning theory and linguistic nativism, see Clark & Lappin (2010b).

[2] See, for example, Laurence & Margolis (2001), Pullum & Scholz (2002), and Crain & Pietroski (2002) for alternative statements of the APS.

and we adopt the computational formulation of this argument given in Clark & Lappin (2010b).

(1) a. Children acquire knowledge of natural language either through domain general learning algorithms or through procedures with strong language specific learning biases that encode the form of a possible grammar.
b. There are no domain general algorithms that could learn natural languages from the primary linguistic data.
c. Children do learn natural languages from primary linguistic data.
d. Therefore children use learning algorithms with strong language specific learning biases that encode the form of a possible grammar.

Some linguists and psychologists have invoked learning theoretic considerations to motivate this version of the APS. So Wexler (1999), apparently referring to some of Gold (1967)'s results, states that

> The strongest most central arguments for innateness thus continue to be the arguments from APS and learnability theory. ... The basic results of the field include the demonstration that without serious constraints on the nature of human grammar, no possible learning mechanism can in fact learn the class of human grammars.

As we will see in Section 3, Gold's results do not entail linguistic nativism. Moreover, his model is highly problematic if taken as a theory of human language learning.

At the other extreme, several linguists have insisted that learning theory has little, if anything of substance to contribute to our understanding of language acquisition. On their approach, we must rely entirely on the empirical insights of psychological and linguistic research in attempting to explain this process. So Yang (2008) maintains that

> In any case, the fundamental problem in language acquisition remains empirical and linguistic, and I don't see any obvious reason to believe that the solution lies in the learning model, be it probabilistic or otherwise.

We suggest that computational learning theory does not motivate strong linguistic nativism, nor is it irrelevant to the task of understanding language acquisition. It will not provide an explanation of this phenomenon. As Yang observes, it is not a substitute for a good psycholinguistic account of the facts. However, it can clarify the class of natural language representations that are efficiently learnable from the PLD. There are a number of important points to keep in mind when considering learning theory as a possible source of insight into language acquisition.

First, as we have already mentioned, a formal learning model is only as good as its basic assumptions concerning the nature of learning, the computational resources with which learners are endowed, and the data available to them. To the extent that these assumptions accurately reflect the situation

of human language learners, the models are informative as mathematical and computational idealizations that indicate the limits of learning in that situation. If they significantly distort important aspects of the human learning context, then the results that they yield will be correspondingly unenlightening in what they tell us about the formal properties of acquisition.

Second, at least some advocates of the APS as an argument for linguistic nativism conflate learnability of the class of natural languages with learnability of a particular grammar formalism.[3] While a formalism may indeed by unlearnable, given reasonable conditions on data, domain general induction procedures, and computational resources, this does not, in itself, show us anything about the learnability of the class of natural languages. In order to motivate an interesting unlearnability claim of the latter sort, it is necessary to show that the formalism in question (or a theory of grammar formulated in this formalism) is the best available representation of the class of natural languages. Establishing such a claim is exceedingly difficult, given that we have yet to achieve even a descriptively adequate grammar for a single language. In its absence, attempting to support the APS on the grounds that a particular grammar formalism is unlearnable from the PLD is vacuous.

Third, it is has often been assumed that the class of natural languages must be identified either with one of the classes in the Chomsky hierarchy of formal languages, or with a class easily definable in terms of this hierarchy.[4] In fact, there is no reason to accept this assumption. As we will see in subsequent sections, there are efficiently learnable classes of languages that run orthogonal to the elements of the Chomsky hierarchy (or are proper subsets of them), and which may be candidates for supersets of the class of natural languages.

Fourth, it is necessary to impose reasonable upper and lower bounds on the degree of difficulty that a learning model imposes on the language learning task. At the lower bound, we want to exclude learning models that trivialize the learning task by neglecting important limitations on the learning process. As we shall see, it is easy to construct models in which almost any class of languages is learnable. Such models are both inaccurate and unhelpful, because they do not constrain or guide our research in any way. At the upper bound we want to avoid theories on which learning is impossibly difficult. Given that humans do achieve the task we seek to model formally, our learning theory must allow for acquisition. If our model does not permit learning, then it is clearly false.

Finally, it is important to distinguish the hypothesis space from which a learning algorithm can select candidate representations of a language, from the

[3] Berwick & Chomsky (2009) identify language acquisition with achieving knowledge of a transformational grammar of a particular kind. See Clark & Lappin (2010b), Chapter 2 for a critical discussion of this and other theory-internal instances of the APS.

[4] See Wintner (2010) for a discussion of the Chomsky hierarchy within formal language theory.

class of languages that it can learn. The learning model imposes constraints that (partially) specify the latter class, but these do not prevent the algorithm from generating hypotheses that fall outside that class. Indeed in some cases it is impossible for the algorithm to restrict its hypotheses so that they lie inside the learnable class. It is also possible for such an algorithm to learn particular languages that are not elements of its learnable class, with particular data sets. Therefore, the class of learnable languages is generally a proper subset of the hypothesis space (hence of the set of representable languages) for a learning algorithm.

It follows that it is not necessary to incorporate a characterization of the learnable class into a language learner as a condition for its learning a specified class of languages. The design of the learner will limit it to the acquisition of a certain class, given data sets of a particular type. However, the design need not specify the learnable class, but only a hypothesis class that might be very much larger than this class.

Moreover, as the set of learnable languages for an algorithm may vary with its input data, this set corresponds to a relational property, rather than to a data invariant feature of the algorithm. In particular, in some models, as the amount of data increases, the class of languages that an algorithm can learn from that quantity of data will also expand. Therefore, only a range of learnable classes of languages, rather than a particular learnable class, can be regarded as intrinsic to the design of a learner.[5]

The tendency to reduce the hypothesis space of a learner to its learnable class runs through the history of the APS, as does the belief that human learners are innately restricted to a narrow class of learnable languages, independently of the PLD to which they are exposed. Neither claim is tenable from a learning theoretic perspective. To the extent that these claims lack independent motivation, they offer no basis for linguistic nativism.

We now turn to a discussion of classical models of learning theory and a critical examination of their defining assumptions. We start with Gold's Identification in the Limit paradigm.

3 Gold's Identification in the Limit Framework

We will take a language to be a set of strings, a subset of the set of all possible strings of finite length whose symbols are drawn from a finite alphabet Σ. We denote the set of all possible strings by Σ^*, and use L to refer to the subset. In keeping with standard practice, we think of the alphabet Σ as the set of words of a language, and the language as the set of all syntactically

[5] See Clark & Lappin (2010b) Chapter 4, Section 7 for a detailed discussion of the relation between the hypothesis space and the learnable class of an algorithm, and for arguments showing why even the specification of the algorithm's learnable class cannot be treated as part of its design.

well-formed (grammatical) sentences. However, the formal results we discuss here apply even under different modeling assumptions. So, for example, we might consider Σ to be the set of phonemes of a natural language, and the language to be the set of strings that satisfy the phonotactic constraints of that language.

Gold (1967)'s *identification in the limit* (IIL) paradigm provides the first application of computational learning theory to the language learning task. In this paradigm a language L consists of a set of strings, and an infinite sequence of these strings is a *presentation* of L. The sequence can be written $s_1, s_2, \ldots$, and every string of a language must appear at least once in the presentation. The learner observes the strings of a presentation one at a time, and on the basis of this evidence, he/she must, at each step, propose a hypothesis for the identity of the language. Given the first string s_1, the learner produces a hypothesis G_1, in response to s_2. He/she will, on the basis of s_1 and s_2, generate G_2, and so on.

For a language L and a presentation of that language $s_1, s_2, \ldots$, the learner identifies in the limit the language L, iff there is some N such that for all $n > N$, $G_n = G_N$, and G_N is a correct representation of L. IIL requires that a learner converge on the correct representation G_L of a language L in a finite but unbounded period of time, on the basis of an unbounded sequence of data samples, and, after constructing G_L, he/she does not depart from it in response to subsequent data. A learner identifies in the limit the class of languages $\mathcal{L}$ iff the learner can identify in the limit every $L \in \mathcal{L}$, for every presentation of strings in the alphabet Σ of L. Questions of learnability concern classes of languages, rather than individual elements of a class.

The strings in a presentation can be selected in any order, so the presentation can be arranged in a way that subverts learning. For example, the first string can recur an unbounded number of times before it is followed by other strings in the language. In order for a class to be learnable in the IIL, it must be possible to learn all of its elements on any presentation of their strings, including those that have been structured in an adversarial manner designed to frustrate learning.

Gold specifies several alternative models within the IIL framework. We will limit our discussion to two of these: the case where the learner receives positive evidence only, and the one where he/she receives both positive and negative evidence.

3.1 The Positive Evidence Only Model

In the positive evidence only variant of IIL presentations consist only of the strings in a language. Gold proves two positive learnability results for this model. Let a *finite language* be one which contains a finite number of strings. This class is clearly infinite, as there are an infinite number of finite subsets of the set of all strings. Gold shows that

(2) **Gold Result 1:**
The class of finite languages is identifiable in the limit on the basis of positive evidence only.

The proof of (2) is straightforward. Gold assumes a rote learning algorithm for this class of languages. When the learner sees a string in a presentation, he/she adds it to the set which specifies the representation of the language iff it has not appeared previously. At point p_i in the presentation, the learner returns as his/her hypothesis G_i = the set of all strings presented up to p_i. If L has k elements, then for any presentation of L, there is a finite point p_N at which every element of L has appeared at least once. At this point G_N will be correct, and it will not change, as no new strings will occur in the presentation.

We can prove a second positive result in this model for any *finite class of languages*. In contrast to the class of finite languages, these classes have a finite number of languages, but may contain infinite languages. We will restrict ourselves throughout this chapter to recursive languages which are defined by the minimal condition that an effective decision procedure exists for deciding membership in the language for any string.

(3) **Gold Result 2:**
A finite class of recursive languages is identifiable in the limit on the basis of positive evidence only.

To prove (3) we invoke a less trivial algorithm than the rote learning procedure used to demonstrate (2). Assume that $\mathcal{L}$ is a finite class of languages, and its elements are ordered by size, so that that if $L_i \subset L_j$, then L_i occurs before L_j. Initially the learning algorithm $\mathcal{A}$ has a list of all possible languages in $\mathcal{L}$, and it returns the first element in that list compatible with the presentation. As $\mathcal{A}$ observes each string s_i in the presentation, it removes from the list all of the languages that do not contain s_i. Eventually it will remove all languages except the correct one L, and the languages that are supersets of L. Given the ordering of the list, $\mathcal{A}$ returns L, the smallest member of the list that is compatible with the presentation, which is the correct hypothesis.

The best known and most influential Gold theorem for the positive evidence only model is a negative result for *supra-finite* classes of languages. Such a class contains all finite languages and at least one infinite language. Gold proves that

(4) **Gold Result 3:**
A supra-finite class of languages is not identifiable in the limit on the basis of positive evidence only.

The proof of (4) consists in generating a contradiction from the assumptions that (i) a class is supra-finite, and (ii) it can be learned in the limit. Take $\mathcal{L}$ to be a supra-finite class of languages, and let $L_{inf} \in \mathcal{L}$ be an infinite language. Suppose that there is an algorithm $\mathcal{A}$ that can identify $\mathcal{L}$ in the

limit. We construct a presentation on which $\mathcal{A}$ fails to converge, which entails that there can be no such $\mathcal{A}$.

Start with the string s_1, where $L_1 = \{s_1\}$ is one of the languages in $\mathcal{L}$. Repeat s_1 until $\mathcal{A}$ starts to produce a representation for L_1 (the presentation will start $s_1, s_1, \ldots$). If $\mathcal{A}$ never predicts L_1, then it will not identify L_1 in the limit, contrary to our assumption. If it does predict L_1, then start generating s_2 until it predicts the finite language $L_2 = \{s_1, s_2\}$. This procedure continues indefinitely, with the presentation $s_1, \ldots, s_2, \ldots, s_3 \ldots$. The number of repetitions of each s_i is sufficiently large to insure that $\mathcal{A}$ generates, at some point, the corresponding language $L_i = \{s_1, \ldots, s_i\}$. This presentation is of the language L_{inf}, which is infinite. But the algorithm will continue predicting ever larger finite subsets of L_{inf} of the form L_i. Therefore, $\mathcal{A}$ will never produce a representation for the infinite language L_{inf}.

Notice that we cannot use the algorithm $\mathcal{A}$ that Gold employs to prove (3) in order to establish that a class of supra-finite languages is identifiable in the limit. This is because a supra-finite class contains the infinite set of all finite languages as a proper subset. If these are ordered in a list by size, and the infinite languages in the class are then ordered as successively larger supersets of the finite elements of this infinite class, then, for any given infinite language L_{inf}, $\mathcal{A}$ will never finish identifying its infinite set of finite language subsets in the list to arrive at L_{inf}.

3.2 The Negative Evidence Model

In Gold's negative evidence (informant) model, a presentation of a language L contains the full set of strings Σ^* generated by the alphabet Σ of L, and each string is labeled for membership either in L, or in its complement L'. Therefore, the learner has access to negative evidence for all non-strings of L in Σ^*. Gold proves that

(5) **Gold Result 4:**
The class of recursive languages is identifiable in the limit in the model in which the learner has access to both positive and negative evidence for each string in a presentation.

Gold proves (5) by specifying an algorithm that identifies in the limit the elements of this class. He takes the enumeration of the class to be an infinite list in which the representations of the language class are ordered without respect to size or computational power. At each point p_i in a presentation the algorithm returns the first representation of a language in the list that is compatible with the data observed up to p_i. This data includes labels for all strings in the sequence $p_1 \ldots p_i$. A representation G_i of a language is compatible with this sequence iff it labels its strings correctly.

The algorithm returns the first G_i in the list that is compatible with the data in the presentation. Because the presentation contains both the strings

of the target language L and the non-strings generated by its alphabet, at some point p_j one of the data samples will rule out all representations in the list that precede G_L, and all samples that follow p_j will be compatible with G_L. Therefore, this algorithm will make only a finite number of errors. The upper bound on the errors that it can make for a presentation corresponds to the integer marking the position of the target representation in the ordered list.

Assume, for example, that L_{fs} is a finite state language which includes the strings of the context-free language L_{cf} as a proper subset. This is the case if $L_{fs} = \{a^n b^m | n, m > 0\}$ and $L_{cf} = \{a^n b^n | n > 0\}$. Let G_{fs} precede G_{cf} in the list of representations for the class. At some point in a presentation for L_{cf} a string labeled as not in the language will appear that is accepted by G_{fs}. As a result, the algorithm will discard G_{fs}, and, by the same process, all other elements of the list, until it arrives at G_{cf}. After this point all data samples will be labeled in accordance with G_{cf}, and so the algorithm will return it. If only positive evidence were contained in the presentation of L_{cf}, all of the data samples would be compatible with G_{fs}, and the algorithm would not be able to identify G_{cf} in the limit.

The class of recursive languages includes the class of context-sensitive languages as a proper subset. To date no natural language has been discovered whose formal syntactic properties exhibit more than context-sensitive resources, and so it seems reasonable to conjecture that natural languages constitute a proper subset of this latter class. Therefore, (5) implies that, with negative evidence for all strings in a language, any natural language can be identified in the limit by the simple learning algorithm that Gold describes.

The negative evidence variant of IIL is an instance in which learning is trivialized by an excessively powerful assumption concerning the sort of evidence that is available to the learner. It is clear that the PLD to which children are exposed does not consist of sentence-label pairs in which every string constructed from the alphabet of the language is identified as grammatical or as ill formed. Whether or not negative evidence of any kind plays a significant role in language acquisition remains a highly controversial issue in psycholinguistics.[6] Even if we assume that certain types of negative evidence are available, it is clear that Gold's full informant model of IIL does not offer a plausible view of the PLD that provides the basis for human language acquisition.

3.3 The Positive Evidence Only Model and Learning Biases

Some linguists have used Gold's proof that a supra-finite class of languages is not identifiable in the limit as grounds for positing a rich set of prior

[6] See Clark & Lappin (2010b), Chapter 3, Section 3.2 for detailed discussion of this issue, as well as Chapter 6 for a proposed stochastic model of indirect negative evidence.

constraints on the human language learning mechanism. So, for example, Matthews (1989) states

> [pp 59-60] The significance of Gold's result becomes apparent if one considers that (i) empiricists assume that there are no constraints on the class of possible natural languages (...), and (ii) Gold's result assumes that the learner employs a maximally powerful learning strategy (...). These two facts ... effectively dispose of the empiricist claim that there exists a "discovery procedure" capable of discovering a grammar for any natural language solely by analyzing a text of that language. This claim can be salvaged but only at the price of abandoning the empiricist program, since one must abandon the assumption that the class of possible languages is relatively unconstrained.

Advocates of linguistic nativism go on to insist that these learning biases must specify the hypothesis space of possible natural languages, and determine a task particular algorithm for selecting elements from this space for given PLD, as necessary conditions for language acquisition. Nowak *et al.* (2001) claim the following.

> Universal grammar consists of (i) a mechanism to generate a search space for all candidate mental grammars and (ii) a learning procedure that specifies how to evaluate the sample sentences. Universal grammar is not learned but is required for language learning. It is innate.

In fact, these conclusions are not well motivated. They depend upon assumptions that are open to serious challenge. First, Gold's negative result concerning supra-finite languages is significant for language acquisition only if one assumes that the class of natural languages is supra-finite, as are the language classes of the Chomsky hierarchy. This need not be the case. A set of languages can be a proper subset of one these classes such that it is a finite class containing infinite languages. In this case, it is not supra-finite, but it is identifiable in the limit. Moreover, it may contain representations that converge on the grammars of natural language.

So, for example, Clark & Eyraud (2007) define the class of substitutable languages, which is a proper subset of the class of context free languages. The grammars of these languages can generate and recognize complex syntactic structures, like relative clauses and polar interrogative questions. Clark & Eyraud (2007) specify a simple algorithm for learning substitutable languages from well formed strings (positive data only). They show that the algorithm identifies in the limit the class of substitutable languages in time polynomial to the required data samples, from a number of samples polynomial bounded by the size of the grammar.

Second, Gold's positive evidence only version of IIL is not a plausible framework for modeling human language acquisition. It is both too demanding of the learner, and too permissive of the resources that it allows him/her. Its excessive rigor consists in the condition that for a class to be identifiable in the limit, all of its elements must be learnable under every presentation.

Therefore, learning is required even when a data presentation is designed in an adversarial mode to sabotage learning. As Gold notes, if we discard this condition and restrict the set of possible presentations to those that promote learning, then we can significantly expand the class of learnable languages, even in the positive evidence only model. Children are not generally subjected to adversarial data conditions, and if they are, learning can be seriously impaired.[7] Therefore, there is no reason to demand learning under every presentation.

Conversely, IIL allows learners unbounded amounts of computational complexity in time and data samples. Identification need only be achieved in the limit, at some bounded point in a presentation. This feature of Gold's framework is unrealistic, given that humans learn under serious restrictions in time, data, and computational power. In order to approximate the human learning process, we need to require that learning be efficient.

Third, as we noted in Section 2, the hypothesis space for a learning algorithm cannot be reduced to the class of representations that it can learn. A grammar induction procedure can generate hypotheses that represent languages outside of its learnable class. It may even learn such languages on particular presentations, but not on all of them.

Finally, the positive evidence only IIL paradigm is too restrictive in requiring exact identification of the target language. Convergence on a particular adult grammar is rarely, if ever, complete. A more realistic approach characterizes learning as a process of probabilistic inference in which the learner attempts to maximize the likelihood of a hypothesis, given the data that it is intended to cover, while seeking to minimize its error rate for this data. We will consider probabilistic learning theories in the next two sections.

4 Probabilistic Models and Realistic Assumptions about Human Learning

One of the limitations of the Gold model is that the learner must identify the target under every possible presentation. Therefore, he/she is required to succeed even when the sequence of examples is selected in order to make the learning task as difficult as possible, ie. even when the teacher is an adversary who is trying to make the learner fail. This is a completely unrealistic view of learning. In the human acquisition process adults generate sentences in the child's environment generally with an interest in facilitating child learning.

A consequence of the IIL is that it is difficult for the learner to tell when a string is not in the language. Absence of evidence in this model is not evidence of absence from the language. The fact that the learner has not seen

[7] Impairment of learning due to an absence of data is particularly clear in the case of feral children, who are deprived of normal linguistic interaction. Perhaps the best known case of such a child is Genie, discussed in Curtiss (1977).

a particular string does not permit him/her to conclude that that string is ill formed. No matter how short a string is, nor how long the learner waits for it, its non-occurrence could be due to the teacher delaying its appearance, rather than ungrammaticality. It is for this reason that, as we have seen, the presence or absence of negative data has such a significant effect on the classes of languages that can be learned within the IIL framework (see Clark & Lappin (2010b) Chapters 3 and 6 for extensive discussion of these issues).

Linguists have been mesmerized by this property of IIL, and they have frequently taken the absence of large amounts of direct negative evidence to be the central fact about language acquisition that motivates the APS (Hornstein & Lightfoot (1981) characterize this issue as the "logical problem of language acquisition"). It is worth noting that it is only in linguistics that the putative absence of negative evidence is considered to be a problem. In other areas of learning it has long been recognised that this is not a particular difficulty. The importance that many linguists assign to negative evidence (more specifically its absence) arises largely because of an unrealistic assumption of the IIL paradigm (Johnson (2004)). From very early on, learning theorists realised that in a more plausible model a learner could infer, from the absence of a particular set of examples, that a grammar should not include some sentences.

(Chomsky, 1981, p. 9) states

> A not unreasonable acquisition system can be devised with the operative principle that if certain structures or rules fail to be exemplified in relatively simple expressions, where they would expect to be found, then a (possibly marked) option is selected excluding them in the grammar, so that a kind of "negative evidence" can be available even without corrections, adverse reactions etc.

This sort of data has traditionally been called "Indirect Negative Evidence". The most natural way to formalise the concept of indirect negative evidence is with probability theory. Under reasonable assumptions, which we discuss below, we can infer from the non-occurrence of a particular sentence in the data that the probability of its being grammatical is very low. It may be that the reason that we have not seen a given example is that we have just been unlucky. The string could actually have quite high probability, but by chance we have not seen it. In fact, it is easy to prove that the likelihood of this situation decreases very rapidly to insignificance. But much more needs to be said. Clearly there are technical problems involved in specifying the relationship between probability of occurrence and grammaticality. First, there are an indefinite number of ungrammatical strings and it is not clear how the learner could keep track of all of these, given his/her limited computational resources.

Second, there are ungrammatical strings that do occur in the PLD. Suppose we have an ungrammatical string with a non-zero probability, say ϵ. Since there are, in most cases, an infinite number of strings in the language, there must be some strings that have probability less than ϵ. In fact, all but finitely

many strings will have probability less than ϵ. This leads to the incovenient fact that the probability of some long grammatical strings will be less than the probability of short ungrammatical ones. Therefore it is clear that we can not simply reduce grammaticality to a particular probability bound.

Returning to the IIL, rather than assuming that the teacher is antagonistic, it seems natural to identify a proper subset as typical or helpful example sequences and require the learner to succeed only on these. It turns out to be difficult to construct a non-trivial model of non-adversarial learning (Goldman & Mathias (1996)). A more realistic approach is to assume that the data has a probabilistic (random) dimension to it. There is much current interest in probabilistic models of language (Bod *et al.* (2003)). We remain neutral as to whether linguistic competence itself should be modeled probabilistically, or categorically as a grammar, with probabilities incorporated into the performance component. Here we are concerned with probabilistic properties of the input data and the learning process, rather than the target that is acquired.

If we move to a probabilistic learning paradigm, then the problem of negative evidence largely disappears. The most basic form of probabilistic learning is *Maximum Likelihood Estimation* (MLE), where we select the model (or set of parameters for a model) that makes the data most likely. When a fixed set of data D (which here corresponds to a sequence of grammatical sentences) is given, the learner chooses an element, from a restricted set of models, that maximises the probability of the data, given that model (this probability value is the likelihood of the model). The MLE approach has an important effect. The smaller the set of strings that the model generates, while still including the data, the higher is its likelihood for that data. To take a trivial example, suppose that there are 5 types of sentences that we could observe, and we see only three of them. A model that assigns a probability of 1/3 to each of the three types that we encounter, and zero probability to the two unseen types, will have higher likelihood than one which gives 1/5 to each of the 5 types. This example illustrates the obvious fact that we do not need explicit negative data to learn that some types do not occur (a point developed more compellingly and more thoroughly in, *inter alia*, Abney (1996); Pereira (2000)).

When we are concerned with cases, as in language acquisition, where there are an unbounded or infinite number of sentence types, it is important to limit the class of models that we can select from. There are many closely related techniques for doing this (like Bayesian model selection and Minimum Description Length), where these techniques enjoy different levels of theoretical support. They all share a common insight. We need to consider not just the likelihood of the model given the data, but we must also take into account the model's size and complexity. Larger and more complex models have to be justified by additional empirical coverage (Goldsmith (2001a)).

In statistical modeling it is standard to regard the data as independently and identically distributed. This it the IID assumption. It entails that for language acquisition there is a fixed distribution over sentences, and each

sentence is chosen randomly from this distribution, with no dependency on the previous example. This claim is clearly false. The distribution of examples does change over time. The relative probabilities of hearing "Good Morning" and "Good Night" depend on the time of day, and there are numerous important inter-sentential dependencies, such as question answer pairs in dialogue.

Many linguists find the IID objectionable for these reasons. In fact, we can defend the IID as an idealization that approximates the facts over large quantities of data. All we need is for the law of large numbers to hold so that the frequency of occurrence of a string will converge to its expected value rapidly. If this is the case, then the effect of the local dependencies among sentences in discourse will be eliminated as the size of the data sample increases. This view of the IID offers a much weaker understanding of the independence conditions than the claim that the sentences of a distribution are generated in full independence of each other. It is a view that applies to a large class of stochastic processes.

Moreover if we can prove learnability under the IID assumption, then we can prove learnability under any other reasonable set of assumptions concerning the distributions of the data as well. Therefore, if we are modeling the acquisition of syntax (i.e. intra-sentential structure), then it is reasonable to neglect the role of inter-sentential dependencies (at least initially). We assume then that there is a fixed distribution. For each string we have a probability. The distribution is just the set of probabilities for all strings in a data set, more accurately, a function that assigns a probability to each string in the set.

To avoid confusion we note that in this chapter we use the word distribution in two entirely different senses. In this section a distribution is a probability distribution over the set of all strings, a function D from $\Sigma^* \rightarrow [0, 1]$, such that the sum over all string of D is equal to 1. In later sections we use distribution in the linguistic sense to refer to the set of environments in which a string can occur.

There are a number of standard models of probabilistic learning that are used in machine learning. The best known of these is the PAC-learning paradigm (Valiant (1984)), where 'PAC' stands for *Probably and Approximately Correct*. The paradigm recognises the fact that if data is selected randomly, then success in learning is random. On occasion the random data that you receive will be inadequate for learning. Unlike the case in IIL, in the PAC framework the learner is not required to learn the target language exactly, but to converge to it probabilistically. This aspect of the paradigm seems particularly well-suited to the task of language learning, but some of its other features rule it out as an appropriate framework for modeling acquisition.

PAC models study learning from labeled data in which each data point is marked for membership or non-membership in the target language. The problem here is, of course, the fact that few, if any, sentences in the PLD are explicitly marked for grammaticality.

A second difficulty is that PAC results rely on the assumption that learning must be (uniformly) possible for all probability distributions over the data. On this assumption, although there is a single fixed distribution, it could be any one in the set of possible distributions. This property of PAC-learning entails that no information can be extracted from the actual probability values assigned to the strings of a language. Any language can receive any probability distribution, and so the primary informational burden of the data is concentrated in the labeling of the strings. The actual human learning context inverts this state of affairs. The data arrives unlabeled, and the primary source of the information that supports learning is the probability distribution that is assigned to the observed strings of the PLD. Therefore, despite its importance in learning theory and the elegance of its formal results, the classical version of PAC-learning has no direct application to the acquisition task. However PAC's convergence measure will be a useful element of a more realistic model.

If we consider further the properties of learnability in the PAC paradigm, we encounter additional problems. A class is PAC learnable if and only if it has a finite VC-dimension, where its VC-dimension is a combinatorial property of the class (see Lappin & Shieber (2007) and Clark & Lappin (2010b), Chapter 5 for characterizations of VC-dimension and discussions of its significance for language learning in the PAC framework). A finite class of languages has finite VC-dimension, and so one way of achieving PAC learnability is to impose a cardinality bound on the target class. So, for example, we might limit the target class to the set of all context-sensitive languages whose description length, when written down, is less than some constant n, the class CS_n. The class of all context-sensitive languages CS has infinite VC-dimension, but we can consider it as the union of a gradually increasing set of classes, $CS = \bigcup_n CS_n$. On the basis of this property of PAC-learning one might be tempted to argue along the following lines for a strong learning bias in language acquisition. As CS has infinite VC-dimension it is not learnable. Therefore the class of languages must be restricted to a member of the set of CS_ns for some n. It follows that language learners must have prior knowledge of the bound n in order to restrict the hypothesis space for grammar induction to the set of CS_ns.

This argument is unsound. In fact a standard result of computational learning theory shows that the learner does not need to know the cardinality bound of the target class. (Haussler *et al.*, 1991). As the amount of available data increases, the learner can gradually expand the set of hypotheses that he/she considers. If the target is in the class CS_n, then the learner will start to consider hypotheses of size n when he/she has access to a sufficiently large amount of data. The size of the hypotheses that he/she constructs grows in proportion to the amount of data he/she observes. A prior cardinality restriction on the hypothesis space is unnecessary.

This point becomes clear when we replace CS with the class of finite languages represented as a list, FIN. A trivial rote learning algorithm can

converge on this class by memorising each observed example for any of its elements. This procedure will learn every element of FIN without requiring prior information on the upper bound for the size of a target language, though FIN has unbounded VC-dimension.

More appropriate learning models yield positive results that show that large classes of languages can be learned, if we restrict the distribution for a language in a reasonable way. One influential line of work looks at the learnability of distributions. On this approach what is learned is not the language itself, but rather the distribution of examples (ie. a stochastic language model).

Angluin (1988) and Chater & Vitányi (2007) extend Horning (1969)'s early work on probabilistic grammatical inference. Their results show that, if we set aside issues of computational complexity, and restrict the set of distributions appropriately, then it is possible to learn classes of grammars that are large enough to include the set of natural languages as a subclass.

As Angluin (1988) says

> These results suggest the presence of probabilistic data largely compensates for the absence of negative data.

Angluin (1988) also considers the learnability of languages under a stochastic version of IIL. She shows, somewhat surprisingly, that Gold's negative results remain in force even in this revised framework. Specifically, she demonstrates that any presentation on which an IIL learner fails can be converted into a special distribution under which a stochastic learner will also not succeed. This result clearly indicates the importance of selecting a realistic set of distributions under which learning is expected. If we require learning even when a distribution is perverse and designed to sabotage acquisition, then we end up with a stochastic learning paradigm that is as implausible as IIL.

The negative results that we derive from either the IIL paradigm or from PAC-learning suffer from an additional important flaw. They do not give us any guide to the class of representations that we should use for the target class, nor do they offer insight into the sort of algorithms that can learn such representations. This is not surprising. Although IIL was originally proposed as a formal model of language acquisition, it quickly became apparent that the framework applies more generally to the task of learning any collection of infinitely many objects. The inductive inference community focuses on learnability of sets of numbers, rather than on sets of strings. Similarly PAC-learning is relevant to every domain of supervised learning. Since these frameworks are not designed specifically for language acquisition, it is to be expected that they have very limited relevance to the construction of a language learning model.

5 Computational Complexity and Efficiency in Language Acquisition

An important constraint on the learner that we have not yet considered is computational complexity. The child learner has limited computational resources and time (a few years) with which to learn his/her language. These conditions impose serious restrictions on the algorithms that the learner can use. These restrictions apply not just to language acquisition, but to other cognitive processes. The Tractable Cognition Thesis (van Rooij (2008)) is uncontroversial.

> Human cognitive capacities are constrained by the fact that humans are finite systems with limited resources for computation.

However, it is not obvious which measure of complexity provides the most appropriate standard for assessing tractability in human computation. Putting aside for a moment the problem of how to formulate the tractability thesis precisely for language acquisition, its consequences are clear. An algorithm that violates this thesis should be rejected as empirically unsound. An inefficient algorithm corresponds to a processing method that a child cannot use, as it requires the ability to perform unrealistic amounts of computation.

It is standard in both computer science and cognitive science to characterise efficient computation as a procedure in which the amount of processing required increases relatively slowly in relation to the growth of an input for a given task. A procedure is generally regarded as tractable if it is bounded by a polynomial function on the size of its input, for the worst processing case. This condition expresses the requirement that computation grow slowly in proportion to the expansion of data, so that it is possible to solve large problems within reasonable limits of time. If the amount of processing that an algorithm $\mathcal{A}$ performs grows very rapidly, by an exponential function on the size of the data, then as the input expands it quickly becomes impossible for $\mathcal{A}$ to compute a result.

Therefore, we can rule out the possibility that child learners use procedures of exponential complexity. Any theory that requires such a procedure for learning is false, and we can set it aside.[8]

We consider the tractability condition to be the most important requirement for a viable computational model of language acquisition to satisfy. The problems involved in efficient construction of a target representation of a language are more substantial than those posed by achieving access to adequate

[8] There are a number of technical problems to do with formalising the idea of efficient computation in this context. For instance, the number data samples that the learner is exposed to increases, and the length of each sample is potentially unbounded. There is no point to restricting the quantity of data that we use at each step in the algorithm, unless we also limit the total size of the data set, and the length of each sample in it.

amounts of data. Efficiency of learning is a very hard problem, and it arises in all learning models, whether or not negative evidence is available.

The computational complexity of learning problems emerges with the least powerful formalisms in the Chomsky hierarchy, the regular languages, and so the more powerful formalisms, like the class of context free (or context sensitive) grammars also suffers from them. These difficulties concern properties of target representations, rather than the language classes as such. It is possible to circumvent some of them by switching to alternative representations which have more tractable learning properties. We will explore this issue in the next section.

There are a number of negative results concerning computational complexity of learning that we will address. Before we do so, we need to register a caveat. All of these results rest on an assumption that a certain class of problem is intrinsically hard to solve. These assumptions, including the famous $P \neq NP$ thesis, are generally held to be true. The results also rely on additional, more obscure presuppositions (such as factoring Blum integers etc.). But these assumptions are not, themselves, proven results, and so we cannot exclude the possibility that efficient algorithms can be devised for at least some of the problems now generally regarded as intractable, although this seems highly unlikely.

The most significant negative complexity results (Gold (1978); Angluin & Kharitonov (1991); Abe & Warmuth (1992); Kearns & Valiant (1994)) show that hard problems can be embedded in the hidden structure of a representation. In particular the results given in Kearns & Valiant (1994) indicate that cryptographically hard problems arise in learning even very simple automata. They entail that the complexity of learning representations is as difficult as code cracking. This suggests that the framework within which these results are obtained does not adequately model human learning. It should distinguish between the supportive environment in which child learners acquire grammar, and the adversarial nature of the code-breaking task. The codes are designed to maximize the difficulty of decryption, while natural languages facilitate acquisition and transmission.

Parametric theories of UG encounter the same complexity issues that other learning models do. Assuming that the hypothesis space of possible grammars is finite does not address the learnabilty issue. In fact, the proofs of the major negative complexity of learning results proceed by defining a series of finitely parameterised sets of grammars, and demonstrating that they are difficult to learn. Therefore, Principles and Parameters (P&P) based models do not solve the complexity problem at the core of the language acquisition task. Some finite hypothesis spaces are efficiently learnable, while others are not. The view that UG consists of a rich set of innate, language specific learning biases that render acquisition tractable contributes nothing of substance to resolving the learning complexity problem, unless a detailed learning model is specified with which efficient learning can be shown. To date, no such model has been offered.

It is important to recognize that the computational hardness of a class of problems hard does not entail that every problem in the class is intractable. It implies only that there are some sets of problems that are hard, and so we cannot construct an algorithm that will solve every problem in the class uniformly. To take a simple example, suppose that the task is clustering. The items that we are presented with are points in a two dimensional plane, and the "language" corresponds to several roughly circular regions. The learning task is to construct a set of clusters of the data where each cluster includes all and only the points with a particular property. Formally this task is computationally hard, since the clusters may contain substantial overlap. If this is the case, then there may be no alternative to trying every possible clustering of the data. However if the clusters are well-separated, the learning task is easy, and it is one that humans perform very well.

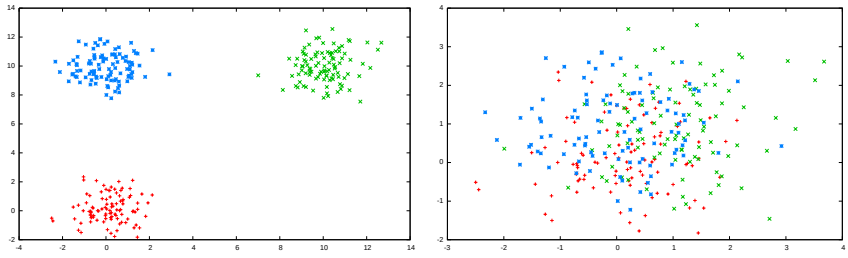

Figure 1. Two clustering problems. On the left the three clusters are well separated and the problem is easy, and on the right they are not, and the problem is hard.

There are provably correct algorithms for identifying clusters that are well-separated, and humans can do this simply by looking at the data on the left of Figure 5. It is easy to draw a circle around each of the three clusters in this diagram. Conversely, when the data are not separated, as in the example on the right of Figure 5, then it is hard to pick out the correct three clusters.

We can represent this difference in hardness by defining a separability parameter. If the centers are well-separated, then the value of the separability parameter will be high, but if they are not, then its value will be low. The parameter allows us to stratify the class of clusters into problems which are easy and those which are hard. Clearly, we do not need to attribute knowledge of this parameter, as a learning prior, to the learner. If the clusters are separated, then the learner will exploit this fact to perform the clustering task, and if they are not, he/she will not succeed in identifying the clusters. From a learnability point of view we could define a class of "learnable clusterings" which are those that are separable. We can prove that an algorithm could learn all of the elements of this class, without incorporating a separability parameter into the algorithm's design.

The analogy between clustering and language learning is clear. Acquiring even simple language representations may be hard in general. However, there might be parameters that divide easy learning problems from hard ones. Stratifying learning tasks in this way permits us use such parameters to identify the class of efficiently learnable languages, and to examine the extent to which natural languages form a subset of this class.

6 Efficient Learning

In fact there are some efficient algorithms for learning classes of representations. Angluin & Kharitonov (1991) shows that there is an important distinction between representations with hidden structure, and those whose structure is more readily discernible from data. Angluin (1987) shows that the class of regular languages can be learned using the class of deterministic finite state automata, when there is a reasonably helpful learning paradigm, but the class of non-deterministic automata is not learnable (Angluin & Kharitonov (1991)). In practice DFAs are quite easy to learn from positive data alone, if this data is not designed to make the learner fail. Subsequent work has established that we can learn DFAs from stochastic data alone, with a helpful distribution on the data set.

If we look at the progress that has been made for induction of DFAs, we see the following stages. First, a simple algorithm is given that can learn a restricted class from positive data alone, within a version of the Gold paraidgm (Angluin (1982)). Next, a more complex algorithm is specified that uses queries or some form of negative evidence to learn a larger set, in this case the entire class of regular languages (Angluin (1987)). Finally, stochastic evidence is substituted for negative data (Carrasco & Oncina (1999)). This sequence suggests that the core issues in learning concern efficient inference from probabilistic data and assumptions. When these are solved, we will be able to model grammar induction from stochastic evidence as a tractable process. The pattern of progress that we have just described for learning theoretic inference of representation classes is now being followed in the modeling of context free grammars induction.

An important question that remains open is whether we will be able to apply the techniques for efficient learning to representation classes that are better able to accommodate natural languages than DFSAs or CFGs. There has been progress towards this goal in recent years, and we will briefly summarize some of this work.

We can gain insight into efficient learnbility by looking at the approaches that have been successful for induction of regular languages. These approaches do not learn just any finite state automaton, but they acquire a finite state

automaton that is uniquely determined by the language. For any regular language L there is a unique minimal DFA that generates it.[9]

In this case, the minimal DFAs, are restricted to only one, and the uniqueness of the device facilitates its learnabiliy. Moreover, they are learnable because the representational primitives of the automaton, its states, correspond to well defined properties of the target language which can be identified from the data. These states are in one-to-one correspondence to what are called the *residual languages* of the language. Given a language L and a string u, the residual language for u, L, written $u^{-1}(L)$ is defined as $\{v|uv \in L\}$. This is just the set of those suffixes of u that form a grammatical string. A well known result, the Myhill-Nerode theorem, establishes that the set of residual languages is finite if and only if the language is regular. In the minimal DFA, each state will generate exactly one of these residual languages.

This DFA has a very particular status. We will call it an *objective* finite automaton. It has the property that the structure of the automaton, though hidden in some sense, is based directly on well defined observable properties of the language that it generates.

Can we specify an analogous *objective* Context Free Grammar with similar learnability properties? There is a class of Deterministic CFGs, but these have the weaker property that the trees which they generate are traversed from left to right. This condition renders an element of the parsing process deterministic, but it does not secure the learnability result that we need.

To get this result we will pursue a connection with the theory of distributional learning, which is closely associated with the work of Zellig Harris (Harris, 1954), and has also been studied extensively by other structuralist linguists (Wells, 1947; Bar-Hillel, 1950). This theory was originally taken to provide discovery procedures for producing the grammar of a language, but it was soon recognized that its techniques could be used to model elements of language acquisition.

The basic concept of distributional learning is, naturally enough, that of a distribution. We define a context to be a sentence with a hole in it, or, equivalently, as a pair of strings (l, r) where l represents the string to the left of the hole, and r represents the one to the right. The distribution of a string u is just the set of contexts in which it can be substituted for the hole to produce a grammatical sentence, and so $C_L(u) = \{(l, r)|lur \in L\}$. Distributional approaches to learning and grammar were studied extensively in the 1950s. One of the clearest expositions is Bar-Hillel (1950), which is largely concerned with the special case where u is a single word. In this instance we are learning only a set of lexical categories.

Joshua Greenberg was another proponent of distributional learning. Chomsky (1959) lucidly paraphrases Greenberg's strategy as "let us say that two

[9] It is possible to relabel the states, but the structure of the automaton is uniquely determined.

units A and B are substitutable$_1$ if there are expressions X and Y such that XAY and XBY are sentences of L; substitutable$_2$ if whenever XAY is a sentence of L then so is XBY and whenever XBY is a sentence of L so is XAY (i.e. A and B are completely mutually substitutable). These are the simplest and most basic notions."

In these terms u is "substitutable$_1$" with v when $C_L(u) \cap C_L(v)$ is non empty and u is "substitutable$_2$" with v when $C_L(u) = C_L(v)$. The latter relation is now called the *syntactic congruence*, and it is easily seen to be an equivalence relation. The equivalence classes for this relation are the congruence classes, expressed as $[u]_L = \{v | C_L(u) = C_L(v)\}$.

It is natural to try to construct an *objective* context free grammar by requiring that the non-terminals of the grammar correspond to these congruence classes, and this approach has yielded the first significant context free grammatical inference result, presented in Clark & Eyraud (2007). Interestingly, the class of CFG languages that this result shows to be learnable is one for which, in Chomsky's terms, one form of substitutability implies the other: a language is substitutable if whenever A and B are substitutable$_1$, then they are substituable$_2$. This class was precisely defined by Myhill in 1950 (Myhill, 1950), which raises the question of why this elementary result was only demonstrated 50 years after the class was first defined. The delay cannot be plausibly attributed to the technical difficulty in the proof of the result in Clark & Eyraud (2007), as this proof is constructed on direct analogy with the proofs given in Angluin (1982).

Rather the difficulty lies in the fact that linguistic theory has been focused on identifying the constituent syntactic structure of a language, which corresponds to the strong generative capacity of a grammar. This structure cannot be uniquely recovered from the PLD without additional constraints on learning. This is because two CFGs may be equivalent in their weak generative power (ie. they generate the same set of strings), but differ in their strong generative capacity (they assign distinct structures to at least some of these strings). Therefore, a learner cannot distinguish between weakly equivalent grammars on he basis of the observed evidence.

In order to achieve the learnability result given in Clark & Eyraud (2007) it is necessary to abandon the idea that grammar induction consists in identifying the correct constituent structure of the language. Instead learning is characterized in terms of recovering the distributional structure of the language. This structure is rich enough to describe the ways in which the primitive units of the language combine to form larger units, and so to specify its syntax, but the resulting grammar, and the parse trees that it produces, do not correspond to the traditional constituents of linguistic theory. This may seem to be a defect of the learning model. In fact it isn't. The constituent structure posited in a particular theory of grammar is itself a theoretical construct invoked to identify the set of grammatical sentences of the language, as speakers represent them. If we can capture these facts through an alterna-

tive representation that is provably learnable, then we have demonstrated the viability of the syntactic structures that this grammar employs.

We have passed over an important question here. We must show show that a learnable grammar is rich enough to support semantic interpretation. We will shortly take up this issue in overview.

In the end, the basic representational assumption of the simple distributional approach is flawed. From a distributional point of view congruence classes give the most fine-grained partitioning of strings into classes that we could devise. Any two strings in a congruence class are fully interchangeable in all contexts, and this condition is rarely, if ever, satisfied. Therefore, a learning algorithm which infers a grammar through identification of these classes will generate representations with large numbers of non-terminals that have very narrow string coverage.

The grammar will also be formally inadequate for capturing the full range of weak generative phenomenon in natural language, because at least some languages contain mildly context sensitive syntactic structures (Shieber, 1985).

Finally, distributional CFGs do not offer an adequate formal basis for semantic interpetation, as neither their tree structures nor their category labels provide the elements of a suitable syntax-semantics interface.

These three considerations indicate that we need a more abstract representation which preserves the learnability properties of the congruence formalism. Our challenge, then, is to combine two putatively incompatible properties: deep, abstract syntactic concepts, and observable, objective structure. It was precisely the apparent conflict between these two requirements that first led Chomsky to discard simple Markov (n-gram) models and adopt linguistic nativism in the form of a strong set of grammar specific learning biases.

In fact there is no intrinsic conflict between the demands of abstract structure on one hand, and categories easily identifiable from the data on the other. Clark (2009) specifies a rich distributional framework that is sufficiently powerful to represent the more abstract general concepts required for natural language syntax, and he demonstrates that this formalism has encouraging learnability properties. It is based on a *Syntactic Concept Lattice*.

The representational primitives of the formalism correspond to sets of strings, but the full congruence of distributional CFGs is replaced by partial sharing of contexts. This weaker condition still generates a very large number of possible categorial primitives, but, by moving to a context-sensitive formalism, we can compute grammars efficiently with these primitives (Clark (2010)). We refer to these representations as *Distributional Lattice Grammars* (DLG), and they have two properties that are important for our discussion of language acquisition.

First, the formalism escapes the limitations that we have noted for simple congruence based approaches. DLGs can represent non-deterministic and inherently ambiguous languages such as

(6) $\{a^n b^n c^m | n, m \geq 0\} \cup \{a^m b^n c^n | n, m \geq 0\}$

It can encode some non-context free languages (such as a variant of the MIX or Bach language), but it cannot represent all context free languages. The examples of context-free languages that the formalism cannot express are artificial, and they do not correspond to syntactic phenomena that are attested in natural languages.

It is important to recognize that our objective here is not to represent the full set of context free grammars, but to model the class of natural languages. It is not a flaw of the DLG framework that it is not able to express some CFGs, if these do not represent natural languages. In fact, this may be taken as a success of the paradigm (Przezdziecki, 2005).

Second, DLGs can be efficiently learned from the data. The current formal results are inadequate in a number of respects. (i) they assume the existence of a membership oracle. The learner is allowed to ask an informant whether a given sentence is grammatical or not. As we discussed above, we consider this to be a reasonable assumption, as long as such queries are restricted in a way that renders them equivalent to indirect negative (stochastic) evidence. (ii) The learnability result is not yet sharp enough. Efficiency is demonstrated for each step in the learning procedure, rather than for the entire process. (iii) Although the formalism exhibits the partial structural completeness that the congruence-based models have, the labels of its parse trees have the rich algebraic structure of a residuated lattice.[10]

The operations in the lattice include the residuation operators / and \, and the partial order in the lattice allows us to define labeled parse trees, where the labels are "maximal" in the lattice. Ambiguous sentences can therefore be assigned sets of different representations, each of which can support a different interpretation. The theory of categorial grammar tells us how we can do this, and Categorial Grammars are based on the same algebraic structure (Lambek (1958)).

The theory of DLGs is still in its infancy, but for the first time we appear to have a learning paradigm that is provably correct, can encode a sufficiently large class of languages, and can produce representations that are rich enough to support semantic interpretation.

The existence of probabilistic data, which we can use as indirect negative evidence, allows us to control for over-generalisation. DLGs provide a very rich framework which can encode the sorts of problems that give rise to the negative results on learning that we have cited. We should not be surprised, then, to find that uniform learning of an entire class in this framework may be hard. So it will certainly be possible to construct combinations of distributions and examples where the learning problem is difficult. But it is crucial to distinguish the assumptions that we make about the learner from those that we adopt for the environment. We can assume that the environment for language

[10] In some circumstances, the derived structural descriptions will not be trees, but non-tree directed acyclic graphs. This will generally be the case when the language is not context-free.

learning is generally benign, but we do not need to attribute knowledge of this fact to the learner.

In the context of the argument from the poverty of the stimulus, we are interested in identifying the minimal initial information which we must assume that the learner has in order to account for acquisition. We are making the following claim for DLGs. In order for acquisition of DLGs to proceed we need to hypothesize a bias for paying attention to the relation between substrings and their contexts, and an ability to construct concept lattices (Ganter & Wille (1997)). The representational formalism and the learning algorithm both follow naturally from these assumptions. Additionally we need to posit a robust mechanism for dealing with noise and sparsity of data. Our second claim is that these mechanisms are adequate for representing a large amount of natural language.

We acknowledge that these claims require substantial empirical support, which has yet to be delivered. We do know that there are a wide range of efficient algorithms for the inference of large classes of context free languages, where these were not available as recently as ten years ago. The exact limits of the approach to learning that we are suggesting have not yet been fully explored. However, the results that we have briefly described here give some reason to think that language acquisition is computationally possible on the basis a set of minimal learning biases. The extent to which these biases are truly domain-general is a subject for future discussion.

7 Machine Learning and Grammar Induction: Some Empirical Results

In the previous sections we have considered the problem of efficient learnability for the class of natural languages from the perspective of formal learning theory. This has involved exploring mathematical properties of learning for different sorts of representation types, under specified conditions of data, time, and computational complexity. In recent years there has been a considerable amount of experimental work on grammar induction from large corpora. This research is of a largely heuristic kind, and it has yielded some interesting results.[11] In this section we will briefly review some of these experiments and discuss their implications for language acquisition.

7.1 Grammar Induction through Supervised Learning

In supervised learning the corpus on which a learning algorithm $\mathcal{A}$ is trained is annotated with the parse structures that are instances of the sort of representations which $\mathcal{A}$ is intended to learn. $\mathcal{A}$ is tested on an unannotated set

[11] For a more detailed discussion of this applied research in grammar induction see Clark & Lappin (2010a).

of examples disjoint from its training set. It is evaluated against the annotated version of the test set, which provides the gold standard for assessing its performance.[12]

$\mathcal{A}$'s parse representations for a test set TS are scored in two dimensions. Its *recall* for TS is the percentage of parse representations from the gold standard annotation of TS that $\mathcal{A}$ returns. $\mathcal{A}$'s *precision* is the percentage of the parse structures that it returns for TS which are in the gold standard. These percentages can be combined as a weighted mean to give $\mathcal{A}$'s F_1-score.[13]

The Penn Treebank (Marcus (1993)) is a corpus of text from the *Wall Street Journal* that has been hand annotated for lexical part of speech (POS) class for its words, and syntactic constituent structure for its sentences. A *Probabilistic Context Free Grammar* (PCFG) is a context-free grammar whose rules are assigned a probability value in which the probability of the sequence of symbols $C_1 \dots C_k$ on the right side of each rule is conditioned on the occurrence of the non-terminal symbol C_0 on the left side, which immediately dominates it in the parse structure. So $P(C_0 \rightarrow C_1 \dots C_k) = P(C_1 \dots C_k | C_0))$.

For every non-terminal C in a PCFG, the probabilities for the rules $C \rightarrow \alpha$ sum to 1. The probability of a derivation of a sequence α from C is the product of the rules applied in the derivation. The probability that the grammar assigns to a string s in a corpus is the sum of the probabilities that the grammar assigns to the derivations for s. The distribution D_G that a PCFG specifies for a language L is the set of probability values that the grammar assigns to the strings in L. If the grammar is consistent, then $\sum_{s \in T^*} D_G(s) = 1$, where T^* is the set of strings generated from T, the set of the grammar's terminal symbols.

The probability values of the rules of a PCFG are its parameters. These can be estimated from a parse annotated corpus by *Maximum Likelihood Estimation* (MLE) (although more reliable techniques for probability estimation are available).

(7) $\frac{c(C_0 \rightarrow C_1 \dots C_k)}{c(C_0 \rightarrow \gamma)}$

where c(R) = the number of occurrences of a rule R in the annotated corpus.

The performance of a PCFG as a supervised grammar learning procedure improves significantly when it is supplemented by lexical head dependencies.

[12] Devising reasonable evaluation methods for natural language processing systems in general, and for grammar induction procedures in particular raises difficult issues. For a discussion of these see Resnik & Lin (2010) and Clark & Lappin (2010a).

[13] Recall, precision, and F-measure were first developed as metrics for evaluating information retrieval and information extraction systems. See Grishman (2010) and Jurafsky & Martin (2009) on their application within NLP.

In a *Lexicalized Probabilistic Context Free Grammar* (LPCFG), the probability of the sequence of symbols on the right side of a CFG rule depends on the pair $\langle C_0, H_0 \rangle$. C_0 is the symbol that immediately dominates the sequence (the left hand side of the rule), and H_0 is the lexical head of the constituent that this symbol encodes, and which the sequence instantiates.

Collins (1999, 2003) constructs a LPCFG that achieves an F-score of approximately 88% for a WSJ test set. Charniak & Johnson (2005) improve on this result with a LPCFG that arrives at an F-score of approximately 91%. This level of performance represents the current state of the art for supervised grammar induction.

Research on supervised learning has made significant progress in the development of accurate parsers for particular domains of text and discourse. However, this work has limited relevance to human language acquisition. The PLD to which children are exposed is not annotated for morphological segmentation, POS classes, or constituent structure. Even if we grant that some negative evidence is contained in the PLD and plays a role in grammar induction, it is not plausible to construe language acquisition as a supervised learning task of the kind described here.

7.2 Unsupervised Grammar Induction

In unsupervised learning the algorithm is trained on a corpus that is not annotated with the structures or features that it is intended to produce for the test set. It must identify its target values on the basis of distributional properties and clustering patterns in the raw training data. There has been considerable success in unsupervised morphological analysis across a variety of languages (Goldsmith (2001b), Goldsmith (2010), Schone & Jurafsky (2001)). Reliable unsupervised POS taggers have also been developed (Schütze (1995), Clark (2003)).

Early experiments on unsupervised parsing did not yield promising results (Carroll & Charniak (1992)). More recent work has produced systems that are starting to converge on the performance of supervised grammar induction. Klein & Manning (2004) (K&M) present an unsupervised parser that combines a constituent structure induction procedure with a head dependency learning method.[14]

K&M's constituent structure induction procedure determines probabilities for all subsequences of POS tagged elements in an input string, where each subsequence is taken as a potential constituent for a parse tree. The procedure invokes a binary branching requirement on all non-terminal elements of the tree. K&M use an *Expectation Maximization* (EM) algorithm to select the parse with the highest probability value. Their procedure identifies (unlabeled)

[14] See Bod (2006, 2007a,b, 2009) for an alternative, largely non-statistical, method of unsupervised parsing.

constituents through the distributional co-occurrence of POS sequences in the same contexts in a corpus. It partially characterizes phrase structure by the condition that sister phrases do not have (non-empty) intersections. Binary branching and the non-overlap requirement are learning biases of the model which the procedure defines.

K&M's unsupervised learning procedure for lexicalized head-dependency grammars assigns probabilities to possible dependency relations in a sentence S. It estimates the likelihood for every word w_i in S that w_i is a head for all of the subsequences of words to its left and to its right, taken as its syntactic arguments or adjuncts. The method computes the likelihood of these alternative dependency relations by evaluating the contexts in which each head occurs. A context consists of the words (word classes) that are immediately adjacent to it on either side. This procedure also imposes a binary branching condition on dependency relations as a learning bias.

K&M combine their dependency and constituent structure grammar systems into an integrated model that computes the score for a constituent tree structure as the product of the values assigned to its terminal elements by the dependency and constituency structure models. This method employs both constituent and head dependency distributional patterns to predict binary constituent parse structure. The method achieves an F-score of 77.6% when it applies to text annotated with Penn Treebank POS tagging, and an F-score of 72.9% when this test set is marked by Schütze (1995)'s unsupervised tagger. The latter case is a more robust instance of unsupervised grammar induction in that the POS tagging on which the learning procedure depends is itself the result of unsupervised word class identification.

7.3 Machine Learning and Language Acquisition

Fong & Berwick (2008) (F&B) argue that supervised parsers, like Collins' LPCFG, do not acquire syntactic knowledge of the sort that characterizes the linguistic competence of native speakers. They run several experiments with variants of Collins' grammar. Their results contain incorrect probabilities for wh-questions, putatively problematic parses for PP attachment cases, and (what they claim to be) some puzzling effects when non-grammatical word order samples are inserted in the data.

Some of the effects that F&B obtain are due to the very limited amount of training data that they employ, and the peculiarities of these samples. It might well be the case that if Collins' LPCFG were trained on a large and suitably annotated subset of the CHILDES child language corpus (MacWinney (1995)), it would yield more appropriate results for the sorts of cases that F&B consider.

But even if their criticisms of Collins' parser are accepted, they do not undermine the relevance of machine learning to language acquisition. As we noted in Section 7.1, supervised learning is not an appropriate model for human learning, because the PLD available to children is not annotated with

target parse structures. Work in unsupervised grammar induction offers more interesting insights into the sorts of linguistic representations that can be acquired from comparatively raw linguistic data through weak bias learning procedures. In order to properly evaluate the significance of this heuristic work for human language acquisition, it is necessary to train and to test machine learning algorithms on the sort of data found in the PLD.

Unsupervised grammar induction is a more difficult task than supervised parsing, and so we might expect F&B's criticisms to apply with even greater force to work in this area. In fact, recent experimental research in unsupervised learning, such as K&M's parsing procedure, indicates that it is possible to achieve accuracy approaching the level of supervised systems. Of course, these results do not show that human language acquisition actually employs these unsupervised algorithms. However, they do provide initial evidence suggesting that weak bias learning methods may well be sufficient to account for language learning. If this is the case, then positing strong biases, rich learning priors, and language specific learning mechanisms requires substantial psychological or neural developmental motivation. The APS does not, in itself, support these devices.

8 Conclusions and Future Research

We have considered the ways in which computational learning theory can contribute insights into language acquisition. We have seen that while formal learning models cannot replace empirically motivated psycholinguistic theories, they can provide important information on the learnability properties of different classes of grammatical representations. However, the usefulness of such models depends on the extent to which their basic assumptions approximate the facts of the human acquisition process.

We looked at two classical learning paradigms, IIL and PAC learning. Each of these has been the source of negative results that linguists have cited in support of the APS. When we examine these results closely we find that they do not, in fact, motivate a strong domain specific bias view of language acquisition. The results generally depend on assumptions that are implausible when applied to acquisition. In some cases, they have been inaccurately interpreted, and, on a precise reading, it becomes clear that they do not entail linguistic nativism.

We observed that the main challenge in developing a tractable algorithm for grammar induction is to constrain the computational complexity involved in inferring a sufficiently rich class of grammatical representations from the PLD. We looked at recent work on probabilistic learning models based on a distributional view of syntax. This line of research has made significant progress in demonstrating the efficient learnability of grammar classes that are beginning to approach the level of expressiveness needed to accommodate natural languages.

A central element in the success of this work is the restriction of the set of possible distributions to those that facilitate learning in a way that corresponds to the PLD to which human learners are exposed. A second important feature is that it characterizes representational classes that are not elements of the Chomsky hierarchy, but run orthogonally to it. A third significant aspect of this work is that although the primitives of the grammars in the learnable classes that it specifies are sufficiently abstract to express interesting syntactic categories and relations, they can be easily identified from the data.

We then considered recent experiments in unsupervised grammar induction from large corpora, where the learning algorithms are of a largely heuristic nature. The results are encouraging, as the unsupervised parsers are beginning to approach the performance of supervised systems of syntactic analysis.

Both the formal and the experimental work on efficient unsupervised grammar induction are in their initial stages of development. Future research in both areas will need to refine the grammar formalisms used in order to provide a fuller and more accurate representation of the syntactic properties of sentences across a larger variety of languages. It is also important to explore the psychological credibility of the learning procedures that successful grammar induction systems employ. This a rich vein of research that holds out the prospect of a rigorously formulated and well motivated computational account of learning in a central human cognitive domain.

References

Abe, N. & M. K. Warmuth (1992), On the computational complexity of approximating distributions by probabilistic automata, *Machine Learning* 9:205–260.

Abney, Steven (1996), Statistical methods and linguistics, in Judith Klavans & Philip Resnik (eds.), *The Balancing Act*, MIT Press, (1–26).

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

Angluin, D. (1987), Learning regular sets from queries and counterexamples, *Information and Computation* 75(2):87–106.

Angluin, D. (1988), Identifying languages from stochastic examples, Technical Report YALEU/ DCS/RR-614, Yale University, Dept. of Computer Science, New Haven, CT.

Angluin, D. & M. Kharitonov (1991), When won't membership queries help?, in *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, ACM New York, NY, USA, (444–454).

Bar-Hillel, Yehoshuah (1950), On syntactical categories, *The Journal of Symbolic Logic* 15(1):1–16.

Berwick, Robert & Noam Chomsky (2009), 'poverty of the stimulus' revisited: Recent challenges reconsidered, in *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, Washington.

Bod, R. (2006), An all-subtrees approach to unsupervised parsing, in *Proceedings of ACL-COLING 2006*, Sydney, Australia, (865–872).

Bod, R. (2007a), Is the end of supervised parsing in sight?, in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, (400–407).

Bod, R. (2007b), A linguistic investigation into unsupervised DOP, in *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, Prague, Czech Republic, (1–8).

Bod, R. (2009), From exemplar to grammar: A probabilistic analogy-based model of language learning, *Cognitive Science* 33:752-793.

Bod, R., J. Hay, & S. Jannedy (2003), *Probabilistic linguistics*, MIT Press.

Carrasco, R. C. & J. Oncina (1999), Learning deterministic regular grammars from stochastic samples in polynomial time, *Theoretical Informatics and Applications* 33(1):1–20.

Carroll, G. & E. Charniak (1992), Two experiments on learning probabilistic dependency grammars from corpora, in C. Weir, S. Abney, R. Grishman, & R. Weischedel (eds.), *Working Notes of the Workshop on Statistically-Based NLP Techniques*, AAAI Press, (1–13).

Charniak, Eugene & Mark Johnson (2005), Coarse-to-fine n-best parsing and maxent discriminative reranking, in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL '05)*, Association for Computational Linguistics, Ann Arbor, Michigan, (173–180).

Chater, N. & P. Vitányi (2007), 'Ideal learning' of natural language: Positive results about learning from positive evidence, *Journal of Mathematical Psychology* 51(3):135–163.

Chomsky, N. (1965), *Aspects of the Theory of Syntax*, MIT Press, Cambridge, MA.

Chomsky, N. (1975), *The Logical Structure of Linguistic Theory*, Plenum Press, New York, NY.

Chomsky, N. (1981), *Lectures on Government and Binding*, Dordrecht: Foris Publications.

Chomsky, N. (1995), *The Minimalist Program*, MIT Press, Cambridge, MA.

Chomsky, N. (2000), *New Horizons in the Study of Language and Mind*, Cambridge University Press, Cambridge.

Chomsky, N. (2005), Three factors in language design, *Linguistic Inquiry* 36:1–22.

Chomsky, Noam (1959), Review of Joshua Greenberg's Essays in Linguistics, *Word* 15:202–218.

Clark, A. & S. Lappin (2010a), Unsupervised learning and grammar induction, in A. Clark, C. Fox, & S. Lappin (eds.), *Handbook of Computational Linguistics and Natural Language Processing*, Wiley-Blackwell, Oxford.

Clark, A. & S. Lappin (2010b), *Linguistic Nativism and the Poverty of the Stimulus*, Wiley-Blackwell, Oxford.

Clark, Alexander (2003), Combining distributional and morphological information for part of speech induction, in *Proceedings of the tenth Annual Meeting of the European Association for Computational Linguistics (EACL)*, (59–66).

Clark, Alexander (2009), A learnable representation for syntax using residuated lattices, in *Proceedings of the conference on Formal Grammar*, Bordeaux, France, to appear.

Clark, Alexander (2010), Efficient, correct, unsupervised learning of context-sensitive languages, in *Proceedings of CoNLL*, Association for Computational Linguistics, Uppsala, Sweden.

Clark, Alexander & Rémi Eyraud (2007), Polynomial identification in the limit of substitutable context-free languages, *Journal of Machine Learning Research* 8:1725–1745.

Collins, M. (1999), *Head-Driven Statistical Models for Natural Language Parsing*, Ph.D. thesis, University of Pennsylvania.

Collins, M. (2003), Head-driven statistical models for natural language parsing, *Computational Linguistics* 29:589–637.

Crain, S. & P. Pietroski (2002), Why language acquisition is a snap, *The Linguistic Review* 18(1-2):163–183.

Curtiss, S. (1977), *Genie: A Psycholinguistic Study of a Modern-day Wild Child*, Academic Press, New York.

Fodor, J.D. & C. Crowther (2002), Understanding stimulus poverty arguments, *The Linguistic Review* 19:105–145.

Fong, S. & R. Berwick (2008), Treebank parsing and knowledge of language: A cognitive perspective, in *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, (539-544).

Ganter, B. & R. Wille (1997), *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag.

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

Gold, E.M. (1978), Complexity of automaton identification from given data, *Information and Control* 37(3):302–320.

Goldman, S.A. & H.D. Mathias (1996), Teaching a Smarter Learner, *Journal of Computer and System Sciences* 52(2):255–267.

Goldsmith, J. (2001a), Unsupervised learning of the morphology of a natural language, *Computational Linguistics* 27(2):153–198.

Goldsmith, J. (2001b), Unsupervised learning of the morphology of a natural language, *Computational Linguistics* 27:153–198.

Goldsmith, J. (2010), Morphology, in A. Clark, C. Fox, & S. Lappin (eds.), *Handbook of Computational Linguistics and Natural Language Processing*, Wiley-Blackwell, Oxford.

Grishman, C. (2010), Information extraction, in A. Clark, C. Fox, & S. Lappin (eds.), *Handbook of Computational Linguistics and Natural Language Processing*, Wiley-Blackwell, Oxford.

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

Haussler, D., M. Kearns, N. Littlestone, & M.K. Warmuth (1991), Equivalence of models for polynomial learnability, *Information and Computation* 95(2):129–161.

Horning, James Jay (1969), *A study of grammatical inference*, Ph.D. thesis, Computer Science Department, Stanford University.

Hornstein, N. & D. Lightfoot (1981), Introduction, in N. Hornstein & D. Lightfoot (eds.), *Explanation in Linguistics: The Logical Problem of Language Acquisition*, Longman, London, (9-31).

Johnson, K. (2004), Gold's Theorem and Cognitive Science, *Philosophy of Science* 71(4):571–592.

Jurafsky, D. & J. Martin (2009), *Speech and Language Processing*, Second Edition, Prentice Hall, Upper Saddle River, NJ.

Kearns, M. & G. Valiant (1994), Cryptographic limitations on learning boolean formulae and finite automata, *JACM* 41(1):67–95.

Klein, D. & Christopher Manning (2004), Corpus-based induction of syntactic structure: Models of dependency and constituency, in *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics*, Barcelnoa, Spain.

Lambek, J. (1958), The mathematics of sentence structure, *American Mathematical Monthly* 65(3):154–170.

Lappin, S. & S. Shieber (2007), Machine learning theory and practice as a source of insight into univeral grammar, *Journal of Linguistics* 43:393–427.

Laurence, S. & E. Margolis (2001), The poverty of the stimulus argument, *British Jounral for the Philosophy of Science* 52:217-276.

MacWinney, B. (1995), *The CHILDES Project: Tools for Analyzing Talk*, Second Edition, Lawrence Erlbaum, Hillsdale, NJ.

Marcus, M. (1993), Building a large annotated corpus of English: The Penn treebank, *Computational Linguistics* 19:313–330.

Matthews, Robert J. (1989), The plausibility of rationalism, in Robert J. Matthews & William Demopoulos (eds.), *Learnability and Linguistic Theory*, Dordrecht, (51–76).

Myhill, John (1950), Review of *On Syntactical Categories* by Yehoshua Bar-Hillel, *The Journal of Symbolic Logic* 15(3):220.

Niyogi, P. & R. C. Berwick (1996), A language learning model for finite parameter spaces, *Cognition* 61:161–193.

Nowak, M. A., N. L. Komarova, & P. Niyogi (2001), Evolution of universal grammar, *Science* 291:114–118.

Pereira, F. (2000), Formal grammar and information theory: Together again?, in *Philosophical Transactions of the Royal Society*, Royal Society, London, (1239–1253).

Pinker, S. (1984), *Language Learnability and Language Development*, Harvard University Press, Cambridge, MA.

Pinker, S. & R. Jackendoff (2005), The faculty of language: What's special about it?, *Cognition* 95:201-236.

Przezdziecki, M.A. (2005), *Vowel harmony and coarticulation in three dialects of Yoruba: phonetics determining phonology*, Ph.D. thesis, Cornell University.

Pullum, G. & B. Scholz (2002), Empirical assessment of stimulus poverty arguments, *The Linguistic Review* 19:9–50.

Resnik, P. & J. Lin (2010), Evaluation of nlp systems, in A. Clark, C. Fox, & S. Lappin (eds.), *Handbook of Computational Linguistics and Natural Language Processing*, Wiley-Blackwell, Oxford.

van Rooij, I. (2008), The tractable cognition thesis, *Cognitive Science: A Multidisciplinary Journal* 32(6):939–984.

Schone, P. & D. Jurafsky (2001), Knowledge-free induction of inflectional morphologies, in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL-2001)*, Pittsburgh, PA.

Schütze, H. (1995), Distributional part-of-speech tagging, in 141-148 (ed.), *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL 7)*.

Shieber, S. (1985), Evidence against the context-freeness of natural language, *Linguistics and Philosophy* 8:333–343.

Valiant, L. (1984), A theory of the learnable, *Communications of the ACM* 27(11):1134 – 1142.

Wells, R.S. (1947), Immediate constituents, *Language* 23(2):81–117.

Wexler, Kenneth (1999), *The MIT Encyclopedia of the Cognitive Sciences*, MIT Press, chapter Innateness of Language, (408–409).

Wintner, S. (2010), Formal language theory, in A. Clark, C. Fox, & S. Lappin (eds.), *Handbook of Computational Linguistics and Natural Language Processing*, Wiley-Blackwell, Oxford.

Yang, C.D. (2002), *Knowledge and Learning in Natural Language*, Oxford University Press, USA.

Yang, Charles (2008), The great number crunch, *Journal of Linguistics* 44(01):205–228.

2 Learnable representations

This paper lays out the basic program: construct representations based on objective properties of the language. This was an invited paper at LATA 2010.

Three learnable models for the description of language

Alexander Clark

Department of Computer Science,
Royal Holloway, University of London
Egham TW20 0EX
alexc@cs.rhul.ac.uk

Abstract. Learnability is a vital property of formal grammars: representation classes should be defined in such a way that they are learnable. One way to build learnable representations is by making them objective or empiricist: the structure of the representation should be based on the structure of the language. Rather than defining a function from representation to language we should start by defining a function from the language to the representation: following this strategy gives classes of representations that are easy to learn. We illustrate this approach with three classes, defined in analogy to the lowest three levels of the Chomsky hierarchy. First, we recall the canonical deterministic finite automaton, where the states of the automaton correspond to the right congruence classes of the language. Secondly, we define context free grammars where the non-terminals of the grammar correspond to the syntactic congruence classes, and where the productions are defined by the syntactic monoid; finally we define a residuated lattice structure from the Galois connection between strings and contexts, which we call the syntactic concept lattice, and base a representation on this, which allows us to define a class of languages that includes some non-context free languages, many context-free languages and all regular languages. All three classes are efficiently learnable under suitable learning paradigms.

1 Introduction

> "Formal language theory was first developed in the mid 1950's in an attempt to develop theories of natural language acquisition."

This statement, in [1], may not be entirely accurate historically, but the point it makes is valid: language theory has its roots in the modelling of learning and of language. The very name of language theory betrays its origins in linguistics. Yet it has moved far from its origins – formal language theory is now an autonomous part of computer science, and only a few papers at the major conferences in Formal Language Theory (FLT) are directly concerned with linguistics. From time to time, the requirements of linguistics impinge on FLT – for example, the discovery of non-context free natural languages [2] inspired much study in

mildly context sensitive grammar formalisms; Minimalist grammars are a direct offshoot of interest in Chomsky's Minimalist Program [3]; and so on.

Learnability is another aspect which has been put to one side. As a model of language acquisition, the original intention was for phrase-structure grammars to be learnable. The PSGs were meant to represent, at a suitable level of abstraction, the linguistic knowledge of the language that the child learner could infer from his or her exposure to the ambient linguistic environment. Chomsky [4] says (p. 172, footnote 15):

> The concept of "phrase structure grammar" was explicitly designed to express the richest system that could reasonable be expected to result from the application of Harris-type procedures to a corpus. ...

The "Harris-type procedures" refer to the methods of distributional learning developed and partially formalised by Zellig Harris [5] following on the ideas of earlier American structuralist linguists. So PSGs in general, and CFGs in particular, were intended, were *designed*, to be learnable by distributional methods — but they weren't. Even the class of regular grammars, equivalent to nondeterministic finite state automata, is not learnable under very easy learning schemes [6]. This is not a problem for distributional methods, but rather a problem with these formalisms. The natural question is therefore whether there are other formalisms, different from the Chomsky hierarchy, that are learnable.

Before we answer this question it is a good idea to be clear about what we mean by learning, and why we think it is so important. Informally learning means that we want to construct our representations for the language from information about the language. From a formal point of view, this means that we wish to define representations, and algorithms for constructing those representations from a source of information about the language, and prove, under a suitable regime, that these algorithms will converge to the right answer. Given a language L, we want to prove that the hypothesised representation will converge to a representation G such that the language defined by G is equal to L. Our focus in this paper is on the computational complexity of learning and so we will assume that we have a very good source of information. We will assume that we have a source of examples of strings from the language, and that additionally the algorithm can query whether a particular string is in the language or not. In the terminology of grammatical inference we have positive data and membership queries. Under some assumptions this is sufficient to learn any class of languages if we neglect computational issues – here we will be considering algorithms that are efficient. For simplicity of exposition we will use a slightly inadequate formalisation: we will merely require that the algorithms use a polynomial amount of computation, in the size of the observed data, at each step of the algorithm.

Of course, not all potential application areas of language theory are linguistic. Linguistics might be considered a special case, where the representations are unknown, as they are cognitive structures and as yet we cannot hope to probe their natures directly. But it is worth considering where the grammars or representations of the languages come from in various other domains. There are broadly speaking two cases: one where we have information about the language,

but not about the representation, and the second is where we have direct information about the representation. In almost all the engineering cases we are interested in, we will have some data that we want to model; in computational biology these might be strings of bases or amino acids, or in other areas they might represent sequences of actions by a robot or a human, or any other sequence of events. But we do not know what the representation is – in none of these cases do we have any direct information about the representation. Indeed, there is no "right" representation, unlike in linguistics. There are merely models that fit the data to a greater or lesser extent.

The only situation where this is not the case is where the language is a programming language or mark up language – in that case, we know what the structure of the language is. In almost all other cases, we will be using languages to represent some sequence of symbols or events, where the representation class is unknown. In these cases, just as in the case of linguistics, learnability is not a tangential property – it is absolutely essential. A representation without some plausible story about how it can be acquired from data is of limited interest. Efficient learning should, in our view, be as important a property of a representation class as efficient parsing.

One way of modelling the data is for a domain expert to construct it by hand. If they are written by a linguist then there are a number of clear desiderata. The grammars should be concise and make it easy for the linguist to express whatever generalisations he or she wishes; it should be easy for humans to reason with consciously, and ideally have a nice graphical representation, perhaps as a tree. Modularity is also a useful property. Following this path we end up with grammatical formalisms being effectively just special-purpose declarative programming languages: DATR [7] and to a lesser extent HPSG [8] are examples of this. There is no reason to think that such formalisms could be or should be learnable. If, on the other hand, we consider the grammar to be the output of a learning process, it need not be particularly convenient for humans to reason about consciously. The properties of a formal grammar are thus radically different, even diametrically opposed depending on whether we envisage them being produced manually or automatically by a process of inference.

2 How

If we accept this argument, then the goal becomes clear – we should construct representations that are intrinsically learnable. Putting this as a slogan: "Put learnability first!" We should design representations from the ground up to be learnable.

Here we present a number of different ways of doing this that follow the same basic strategy: they are objective or "empiricist". We define the representational primitives of the formalism in a language theoretical way. The basic elements of the formalism, whether they are states in an automaton, or non-terminals in a phrase structure grammar, must have a clear definition in terms of sets of strings, in a way that does not depend on the representation.

Rather than defining a representation, and then defining a function from the representation to the language, we should go backwards. If we are interested first in learnability, then we should start by defining the map from the language to the representation. For example, in a CFG, we define a derivation relation $\stackrel{*}{\Rightarrow}$; for each non-terminal N we define the set of strings that can be derived from that non-terminal: $Y(N) = \{w | N \stackrel{*}{\Rightarrow} w\}$. We then define the language as the set of strings derived from the start symbol or symbols. Thus we define a function from a non-terminal N, to a set of strings $Y(N)$, and thus from the set of context free grammars to the set of context free languages: from G to $L(G)$.

There is however an insurmountable obstacle to going in the reverse direction: $Y(N)$ is almost completely unconstrained. Suppose we have some context free language L, and a grammar G such that $L(G) = L$ and N is a non-terminal in G. What constraints are there on $Y(N)$? We can say literally nothing about this set, other than that it is a context free language. If we restrict the CFG so that all non-terminals are accessible, then it must be a subset of the set of substrings of L, but beyond that we can't say anything. Consider for example the degenerate case when L is Σ^*. There are clearly many CFGs with one non-terminal that define this language, but there are also infinitely many other ones, with non-terminals that correspond to arbitrary context-free subsets of Σ^*. We need some way of identifying some relevant subsets that will correspond to the primitive elements of our representation. We should start by defining some suitable sets of strings; only then can we construct a grammar. If we define some set of strings then the structure of the representation will follow: given an objective definition of the "reference" of the non-terminal or symbol, the derivation process will be fixed.

We will define three representations; we will start with a basic representation that is quite standard, and turns out to be equivalent to a subclass of DFAs; indeed to the canonical automata. In this case we base the representation on the right congruence classes, as used in the Myhill-Nerode theorem. The next step is to define context free grammars where the non-terminals correspond to the syntactic congruence classes of the language; the third and final representation uses what we call the syntactic concepts of the language – elements of a residuated lattice – as the representational primitives, and the resulting formalism can represent some non-context-free languages.

3 Canonical DFA

We will start by considering a classic case [9–11], where the learnability problems are well understood: the case of regular languages. We will end up with a class of representations equivalent to a subclass of deterministic finite automata. We present this standard theory in a slightly non-standard way in order to make the step to the next class of representations as painless as possible.

We will define our notation as needed; we take a finite non-empty aphabet Σ, and the free monoid Σ^*; we use λ to refer to the empty string. A language is a subset of Σ^*. Let L be some arbitrary language; not necessarily regular,

or even computable. We define the residual language of a given string u as $u^{-1}L = \{w : uw \in L\}$.

Consider the following relation between strings: $u \sim_L v$ iff $u^{-1}L = v^{-1}L$. This is an equivalence relation and additionally a right congruence: if $u \sim_L v$ then for all $w \in \Sigma^*$, $uw \sim_L vw$.

We can consider the equivalence classes under this relation: we will write $[u]^R$ for the congruence class of the string u under this right congruence. It is better to consider these classes not just as sets of strings but as pairs $\langle P, S\rangle$ where P is a congruence class, and S is the residual language of all the strings in P. That is to say we will have elements of the form $\langle [u]^R, u^{-1}L\rangle$. One important such element is $\langle [\lambda]^R, L\rangle$.

Suppose we define a representation that is based on these congruence classes. Let us call these primitive elements of our representation *states*. The state $\langle [\lambda]^R, L\rangle$ we will denote by q_0. A few elementary observations: if $u \in L$ then every element of $[u]^R$ is also in L. We will call a state $\langle P, S\rangle$ such that $\lambda \in S$ a *final* state. Thus if we can tell for each string in the language which congruence class it is in, then we will have predicted the language. Our representation will be based on this idea: we will try to compute for each string w not just whether it is in L but which congruence class it is in.

We have now taken the first step: we have defined the primitive elements of the representation. We now need to define a derivation of some sort by exploiting the algebraic structure of these classes, in particular the fact that they are right congruence classes. Since it is a right congruence, if we have a string that we know is in the congruence class $[u]^R$ and we append the string v we know that it will be in the class $[uv]^R$. Thus we have a "transition" from the state $[u]^R$ to the state $[uv]^R$ labelled with the string v. It is clear that we can restrict ourselves to the case where $|v| = 1$, i.e. where the transitions are labelled with letters. We now have something that looks very like an automaton. We let Q be the, possibly infinite set of all these states, q_0 the initial state, δ a transition function defined by $\delta([u]^R, a) = [ua]^R$, and let F be the set of final states $\{[u]^R | u \in L\}$. We now define $\mathfrak{R}(L)$ to be this, possibly infinite, representation. We have defined a function from L to $\mathfrak{R}(L)$.

We extend δ recursively in the standard way and then define the function from the representation to the language $L(\mathfrak{R}(L)) = \{w : \delta(q_0, w) \in F\}$. Given this, we have that for any language L, $L(\mathfrak{R}(L)) = L$. In a strict sense, this "representation" is correct. Of course we are interested in those cases where we have a finite representation, and $\mathfrak{R}(L)$ will be finite if and only if L is regular, by the Myhill-Nerode theorem. Thus while we have that for any language this representation is correct, we can only finitely represent the class of regular languages.

It is possible to infer these representations for regular languages, using a number of different techniques depending on the details of the source of information that one has about the language. The recipe is then as follows:

- Define a set of primitive elements language theoretically – in this case the right congruence classes: $[u]^R$.

– Identify a derivation relation of some sort based on the algebraic structure of these elements: $[u]^R \rightarrow^a [v]^R$.
– Construct a representation $\mathfrak{R}(L)$ based on the language and prove that it is correct $L(\mathfrak{R}(L)) = L$.
– Identify the class of languages that can be finitely represented in this way: the class of regular languages. Happily, in this case, it coincides with an existing class of languages.

4 CFGs with congruence classes

We can now move on from representations that are regular to ones that are capable of representing context-free languages. We do this using the idea of distributional learning. These techniques were originally described by structuralist linguists who used them to devise mechanical procedures for discovering the structure of natural languages. As such, they are a reasonable starting point for investigations of learnable representations.

Some notation and basic terminology: we define a context to be a pair of strings (l, r) where $l, r \in \Sigma^*$. We combine a context with a substring so $(l, r) \odot u = lur$; We will sometimes refer to a context (l, r) with a single letter f. A string u occurs in a context (l, r) in a language if $lur \in L$. If L, R are sets of strings then we use (L, R), (L, r) etc to refer to the obvious sets of contexts: $L \times R$, $L \times \{r\}$ and so on. We define the distribution of a string in a language as the set of all contexts that it can occur in: $C_L(w) = \{(l, r) | lur \in L\}$. We will extend the notation $\odot$ to contexts: $(l, r) \odot (x, y) = (lx, yr)$, so $(f \odot g) \odot u = f \odot (g \odot u)$. We will also use it for sets in the obvious way.

We now define the crucial notion for this approach: two strings u and v are syntactically congruent iff they have the same distribution: $u \equiv_L v$, iff $C_L(u) = C_L(v)$. We write $[u]$ for the congruence class of u. We note here a classic result – the number of congruence classes is finite if and only if the language is regular.

Given the discussion in the previous section we hope that it is obvious what the next step is: our primitive elements will correspond to these congruence classes. Immediately this seems to raise the problem that we will be restricted to regular languages, since we are interested in finite representations, and thus can only represent a finite number of congruence classes. This turns out not to be the case, as we shall see.

Clearly the empty context (λ, λ) has a special significance: this context is in the distribution of a string if and only if that string is in the language; $(\lambda, \lambda) \in C_L(u)$ means that $u \in L$. So if we can predict the congruence class of a string, we will know whether the string is in the language or not. Given this fixed interpretation of these symbols, we can now proceed to determine what the appropriate derivation rules should be. We have mentioned that this is a congruence: this means that for all strings u, v, x, y if $u \equiv_L v$ then $xuy \equiv xvy$. This means that if we take any element of $[u]$ say u' and any element of $[v]$ say v', and concatenate them, then the result $u'v'$ will always be in the same congruence class as $[uv]$. This means that if we want to generate an element of

$[uv]$ we can do this by generating an element from $[u]$ and then generating an element from $[v]$ and then concatenating the results. In other words, we have a context free production $[uv] \to [u][v]$. Additionally we know that for any string w we can generate an element of $[w]$ just by producing the string w. Given the previous productions, it is clearly sufficient just to have these productions for strings of length 1: i.e. to have productions $[a] \to a$.

Another way of viewing this is to note that the concatenation of the congruence classes is well defined – or alternatively that since the relation $\equiv_L$ is a monoid congruence, we can use the quotient monoid $\Sigma^* / \equiv_L$ which is the well known syntactic monoid. The production rules then can be viewed as saying that $X \to YZ$ is a production if and only if $X = Y \circ Z$, and that $X \to a$ iff $a \in X$. Here, we can see even more clearly that the structure of the representation is based on the structure of the language – in this case we have a CFG-like formalism that corresponds exactly to the syntactic monoid of the language. We therefore define our representation as follows: $\mathfrak{C}(L)$ consists of the, possibly infinite, set of congruence classes $[u]$ together with a set of productions consisting of $\{[uv] \to [u], [v] | u, v \in \Sigma^*\}$ and $\{[a] \to a | a \in \Sigma\}$ and $[\lambda] \to \lambda$. We identify a set of initial symbols $I = \{[u] | u \in L\}$, and we define derivation exactly as in a context free grammar.

It is then easy to see that given these productions, $[w] \stackrel{*}{\Rightarrow} v$ iff $v \in [w]$. Thus the productions are well behaved. We then define $L(\mathfrak{C}(L)) = \{w | \exists N \in I$ such that $N \stackrel{*}{\Rightarrow} w\}$. We can then prove that $L(\mathfrak{C}(L)) = L$, for any language L.

We have used the two schemas $[uv] \to [u][v]$ and $[a] \to a$; these are sufficient. But it is conceivable that we might want to have different schemas – we can have schemas like $[w] \to w$, $[lwr] \to l[w]r$, $[aw] \to a[w]$ or even $[uvw] \to [u][v][w]$, which will give us finite grammars, linear grammars, regular grammars and so on. All of these schemas maintain the basic invariant that they will only derive strings of the same congruence class.

We thus end up with something that looks something like a context free grammar in Chomsky normal form. It differs in two respects, one trivial and one extremely important. The trivial difference is that we may have more than one start symbol: we wish to maintain the nice map between the representation and the structure.

The second point is that the number of congruence classes will be infinite, if the language is not regular. Consider the language $L_{ab} = \{a^n b^n | n \geq 0\}$. This is a non-regular context free language. It is easy to see that we have an infinite number of congruence classes since a^i is not congruent to a^j unless $i = j$. It appears therefore that we have only achieved another representation for regular languages. We can consider this as the *canonical context free grammar* for a regular language.

Let us suppose we maintain the structure of the representation but only take a finite set of congruence classes V consisting of the classes corresponding to a finite set of strings K: $V = \{[u] | u \in K\}$. We will assume that K contains Σ and λ and finitely many others strings. The binary productions will thus be limited to the finite set $\{[uv] \to [u][v] | [u], [v], [uv] \in V\}$. This will then give us a finite

representation, which we denote $\mathfrak{C}(L, K)$ We can prove that if we have only a subset of the productions, then $[w] \stackrel{*}{\Rightarrow} v$ implies $v \in [w]$, and therefore our representation will always generate a subset of the correct language: $L(\mathfrak{C}(L, K)) \subseteq L$.

The class that we can represent is therefore the set of all languages L such that there is some finite set of strings K such that $\mathfrak{C}(L, K)$ defines the correct language.

$$\mathcal{L}_{\text{CCFG}} = \{L | \exists \text{ finite } K \subset \Sigma^* \text{ such that } L(\mathfrak{C}(L, K)) = L\}$$

This class clearly includes the class of regular languages. It also includes some non-regular context free languages. In the case of our example L_{ab} it is sufficient to have the following congruence classes: $[a], [b], [\lambda], [ab], [aab], [abb]$. Not all context free languages can be described in this way. The context free language $\{a^n b^m | n < m\}$ is not in $\mathcal{L}_{\text{CCFG}}$, as the language is the union of an infinite number of congruence classes. $\mathcal{L}_{\text{CCFG}}$ is therefore a proper subclass of the class of context free languages; by restricting the non-terminals to correspond exactly to the congruence classes, we lose a bit of representational power, but we gain efficient learnability. Note the very close relationship to the class of NTS languages [12]; indeed we conjecture that these classes may be equal. The first results on learning using this approach [13–15] have shown that various subclasses can be learned under various non-probabilistic and probabilistic paradigms. Note that we have lost the canonical nature of the formalism – there will often be more than one possible minimal choice of K or V. Nonetheless, given that the definition of the primitives is fixed this is not a problem: any sufficiently large set of congruence classes will suffice to define the same language.

4.1 Regular languages

Let us briefly return to the case of regular languages. We know that the set of congruence classes is finite, but we can get some insight into the structure of this set by looking at the proof. Let A be the minimal deterministic finite automaton for a language L. Let Q be the set of states of this automaton; let $n = |Q|$. A string w defines a function from Q to Q: $f_w(q) = \delta(q, w)$. Clearly there are only n^n possible such functions. But if $f_u = f_v$ then $u \equiv_L v$, and so there can be at most n^n possible congruence classes. Indeed Holzer and Konig [16] show that we can approach this bound. This reveals two things: one that using one non-terminal per congruence class could be an expensive mistake as the number might be prohibitively large, and secondly, that there is often some non-trivial structure to the monoid.

Since these congruence classes correspond to functions from Q to Q it seems reasonable to represent them using some basis functions. Consider the set of partial functions that take $q_i \to q_j$: there are only n^2 of these. Each congruence class can be represented as a collection of at most n of these that define the image under f of each $q \in Q$.

If we represent each congruence class as a n by n boolean matrix T; where T_{ij} is 1 iff $f_u : q_i \mapsto q_j$, then the basis functions are the n^2 matrices that have

just a single 1; and we represent each congruence class as a sum of n such basis functions.

Suppose the language is reversible [17]: i.e. $uv, u'v, uv' \in L$ implies $u'v' \in L$. Then for each state q_i we can define a pair of strings l_i, r_i that uniquely pick out that state: in the sense that $\delta(q_0, l_i) = q_i$ and only q_i has the property that $\delta(q_i, r_i)$ is a final state.

Thus we can represent the basis functions using the finite set of contexts (l_i, r_j) that represents the transition function $q_i \to q_j$. This gives us an important clue how to represent the syntactic monoid: If we have a class that maps $q_i \to q_j$ and another which maps $q_j \to q_k$ then the concatenation will map $q_i \to q_k$. Thus rather than having a very large number of very specific rules that show how individual congruence classes combine, we can have a very much smaller set of more general rules which should be easier to learn. This requires a representation that contains elements that correspond not just to individual congruence classes but to sets of congruence classes.

5 Distributional lattice grammars

The final class of representations that we consider are based on a richer algebraic structure; see [18] for a more detailed exposition. Note that the congruence classes correspond to sets of strings and dually to sets of contexts: a congruence class $[u]$ also defines the distribution $C_L(u)$ and vice versa. It is natural to consider therefore as our primitive elements certain ordered pairs which we write $\langle S, C \rangle$ where S is a subset of Σ^* and C is a subset of $\Sigma^* \times \Sigma^*$. Given a language L we will consider only those pairs that satisfy two conditions: first that $C \odot S$ is a subset of L, and secondly that both of these sets are maximal, while respecting the first condition. If a pair satisfies these conditions, then we call it a *syntactic concept* of the language.

We have chosen to define it in this way to bring out the connection to the Universal automaton [19, 20], which has the same construction but using a prefix-suffix relation, rather than the context substring relation.

Another way is to consider the Galois connection between the sets of strings and contexts, which give rise to exactly the same sets of concepts. For a given language L we can define two polar maps from sets of strings to sets of contexts and vice versa. Given a set of strings S we can define a set of contexts S' to be the set of contexts that appear with every element of S.

$$S' = \{(l, r) : \forall w \in S \;\; lwr \in L\} \tag{1}$$

Dually we can define for a set of contexts C the set of strings C' that occur with all of the elements of C

$$C' = \{w : \forall (l, r) \in C \;\; lwr \in L\} \tag{2}$$

A concept is then an ordered pair $\langle S, C \rangle$ where $S' = C$ and $C' = S$. The most important point here is that these are closure operations in the sense that

$S''' = S'$ and $C''' = C'$. This means that we can construct a concept by taking any set of strings S and computing $\langle S'', S' \rangle$, and similarly by taking any set of contexts C and computing $\langle C', C'' \rangle$. We will write $\mathcal{C}(S)$ for $\langle S'', S' \rangle$. This set of concepts have an interesting and rich algebraic structure, which gives rise to a powerful representation.

We will start by stating some basic properties of the set of concepts. The first point is that there is an inverse relation between the size of the set of strings S and the set of contexts C: the larger that S is the smaller that C is: in the limit there is always a concept where $S = \Sigma^*$; normally this will have $C = \emptyset$. Conversely we will always have an element $\mathcal{C}(\Sigma^* \times \Sigma^*)$. One particularly important concept is $\mathcal{C}(L) = \mathcal{C}((\lambda, \lambda))$: the language itself is one of the concepts.

The most basic structure that this set of concepts has is therefore as a partially ordered set. We can define a partial order on these concepts where:

$$\langle S_1, C_1 \rangle \leq \langle S_2, C_2 \rangle \text{ iff } S_1 \subseteq S_2.$$

$S_1 \subseteq S_2$ iff $C_1 \supseteq C_2$. We can see that $\mathcal{C}(L) = \mathcal{C}(\{(\lambda, \lambda)\})$, and clearly $w \in L$ iff $\mathcal{C}(\{w\}) \leq \mathcal{C}(\{(\lambda, \lambda)\})$.

Given this partial order we can see easily that in fact this forms a complete lattice; which we write $\mathfrak{B}(L)$, called the *syntactic concept lattice*. Here the topmost element is $\top = \mathcal{C}(\Sigma^*)$ bottom is written $\bot = \mathcal{C}(\Sigma^* \times \Sigma^*)$, and the two meet and join operations are defined as $\langle S_x, C_x \rangle \wedge \langle S_y, C_y \rangle$ is defined as $\langle S_x \cap S_y, (S_x \cap S_y)' \rangle$ and $\vee$ dually as $\langle (C_x \cap C_y)', C_x \cap C_y \rangle$.

Figure 1 shows the syntactic concept lattice for the regular language $L = \{(ab)^*\}$. L is infinite, but the lattice $\mathfrak{B}(L)$ is finite and has only 7 concepts.

Monoid structure

Crucially, this lattice structure also has a monoid structure. We can define a binary operation over concepts using the two sets of strings of the concepts: define $\langle S_1, C_1 \rangle \circ \langle S_2, C_2 \rangle = \mathcal{C}(S_1 S_2)$. Note that this operation then forms a monoid, as it is both associative and has a unit $\mathcal{C}(\lambda)$. There is also an interaction between this monoid structure and the lattice structure. It is clearly monotonic in the sense that if $X \leq Y$ then $X \circ Z \leq Y \circ Z$ and so on, but there is a stronger relation. We can define two residual operations as follows:

Definition 1. *Suppose* $X = \langle S_x, C_x \rangle$ *and* $Y = \langle S_y, C_y \rangle$ *are concepts. Then define the residual* $X/Y = \mathcal{C}(C_x \odot (\lambda, S_y))$ *and* $Y \backslash X = \mathcal{C}(C_x \odot (S_y, \lambda))$

These satisfy the following conditions: $Y \leq X \backslash Z$ iff $X \circ Y \leq Z$ iff $X \leq Z/Y$. That is to say, given an element Z and an element X, $X \backslash Z$ is the largest element which when concatenated to the right of X will give you something that is less than Z.

With these operations the syntactic concept lattice becomes a residuated lattice. This gives some intriguing links to the theory of categorial grammars [21].

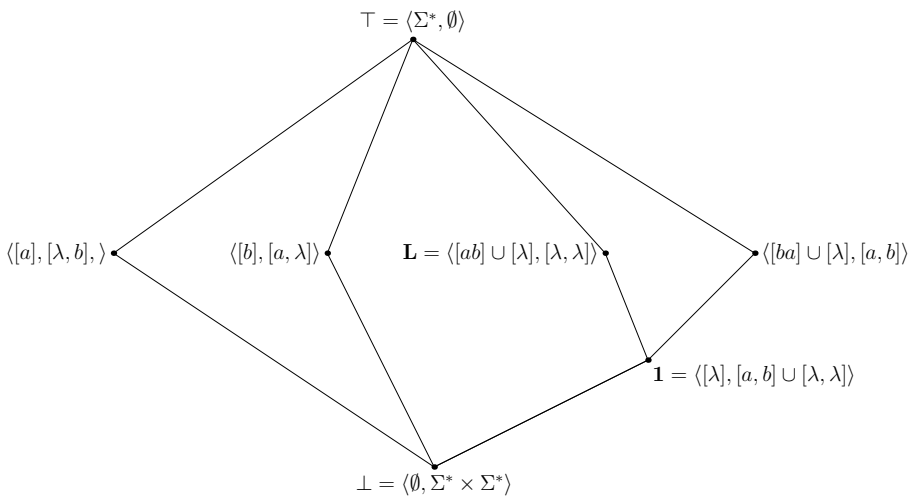

Fig. 1. The Hasse diagram for the syntactic concept lattice for the regular language $L = \{(ab)^*\}$. Each concept (node in the diagram) is an ordered pair of a set of strings, and a set of contexts. We write $[u]$ for the equivalence class of the string u, $[l, r]$ for the equivalence class of the context (l, r), under equality of

5.1 Maximal elements

One important point of the residual operation is that we can use them to extract maximally general concatenation rules. So, suppose we have some concept Z. We can consider the set of all pairs of concepts (X, Y) such that their concatenation is less than Z.

$$H(Z) = \{(X, Y) | X \circ Y \leq Z\} \tag{3}$$

This is clearly a down set, in that if $(X, Y) \in H(Z)$ and $X' \leq X$ and $Y' \leq Y$ then $(X', Y') \in H(Z)$, and so it is natural to consider the maximal elements of this set. We can find these elements using the residuation operations. If $(X, Y) \in H(Z)$ then also we must have $(X, X \backslash Z)$ and $(Z/Y, Y)$ in $H(Z)$. But we can repeat this: we will also have $(Z/(X \backslash Z), X \backslash Z)$ and $(Z/Y, (Z/Y) \backslash Z)$ in $H(Z)$. We can prove that repeating the process further is not necessary – indeed all maximal elements of $H(Z)$ will be of this form.

An example: suppose $L = \{a^n b^n | n \geq 0\}$. Consider $H(\mathcal{C}(L))$. Clearly $\mathcal{C}(a) \circ \mathcal{C}(b) = \mathcal{C}(L)$, so $(\mathcal{C}(a), \mathcal{C}(b)) \in H(\mathcal{C}(L))$. Let us identify the two maximal elements that we get by generalising this pair. $\mathcal{C}(L)/\mathcal{C}(b) = \mathcal{C}(aab)$, and $\mathcal{C}(a) \backslash \mathcal{C}(L) = \mathcal{C}(abb)$. Repeating the process does not increase these so the two maximal elements above this pair are $(\mathcal{C}(a), \mathcal{C}(abb))$ and $(\mathcal{C}(aab), \mathcal{C}(b))$.

6 Representation

Having defined and examined the syntactic concept lattice, we can now define a representation based on this. Again, since the lattice will be infinite if the language is not regular, we need to consider just a part of it. We will start by considering how we might define a representation given the whole lattice. Given a string w we want to compute whether it is in the language or not. Considering this slightly more generally we want to be able to compute for every string w, the concept of w, $\mathcal{C}(w)$. If $\mathcal{C}(w) \leq \mathcal{C}(L)$, then we know that the string is in the language. If we have the whole lattice then it is quite easy: since $\mathcal{C}(u) \circ \mathcal{C}(v) = \mathcal{C}(uv)$, we can simply take the list of letters that form w and concatenate their concepts. So if $w = a_1 \dots a_n$, then $\mathcal{C}(w) = \mathcal{C}(a_1) \circ \dots \circ \mathcal{C}(a_n)$. It is enough to know the concepts of the letters, and of course of λ and the algebraic operation $\circ$ which is associative and well-behaved. In this case it effectively reduces to computing the syntactic monoid as before. The idea is very simple – we compute a representation of the distribution of a string, by taking the distributions of its parts and combining them.

However, if we have a non-regular language, then we will need to restrict the lattice in some way. We can do this by taking a finite set of contexts $F \subset \Sigma^* \times \Sigma^*$, which will include the special context (λ, λ) and constructing a lattice using only these contexts and all strings Σ^*. This give us a finite lattice $\mathfrak{B}(L, F)$, which will have at most $2^{|F|}$ elements. We can think of F as being a set of *features*, where a string w has the feature (context) (l, r) iff $lwr \in L$.

Definition 2. *For a language L and a set of context $F \subseteq \Sigma^* \times \Sigma^*$, the partial lattice $\mathfrak{B}(L, F)$ is the lattice of concepts $\langle S, C \rangle$ where $C \subseteq F$, and where $C = S' \cap F$, and $S = C'$.*

We can define a concatenation operation as before

$$\langle S_1, C_1 \rangle \circ \langle S_2, C_2 \rangle = \langle ((S_1 S_2)' \cap F)', (S_1 S_2)' \cap F \rangle$$

This is now however no longer a residuated lattice as the $\circ$ operation is no longer associative, there may not be an identity element, nor are the residuation operations well defined. We will now clearly not be able to compute things exactly for every language, but we should still be able to approximate the computation, and for some languages, and for some sets of features the approximation will be accurate.

We note some basic facts about this partial lattice: first it is no longer the case that $\mathcal{C}(u) \circ \mathcal{C}(v) = \mathcal{C}(uv)$. If we take a very long string in the language, we may be able to split it into two parts neither of which have any contexts ing F. For example, if we have the language of our running example $a^n b^n$, and a small set of short contexts, we could take the string $a^{100} b^{100}$ and split it into the two strings a^{100} and b^{100}. Neither of these two strings will have any contexts in F, and so $\mathcal{C}(a^{100}) = \mathcal{C}(b^{100}) = \top$; this means that $\mathcal{C}(a^{100}) \circ \mathcal{C}(b^{100}) = \top \circ \top = \top > \mathcal{C}(a^{100} b^{100}) = \mathcal{C}(L)$: they are not equal.

However we can prove that $\mathcal{C}(u) \circ \mathcal{C}(v) \geq \mathcal{C}(uv)$. This means that given some string, w, we can compute an upper bound on the $\mathcal{C}(w)$ quite easily: we will call this upper bound $\phi(w)$. This may not give us exactly the right answer but it will still be useful. If the upper bound is below $\mathcal{C}(L)$ then we definitely know that the string will be in the language: if $\mathcal{C}(w) \leq \phi(w)$ and $\phi(w) \leq \mathcal{C}(L)$, then $\mathcal{C}(w) \leq \mathcal{C}(L)$.

In fact we can compute many different upper bounds: since the operation is no longer associative, the order in which we do the computations matters. Suppose we have some string ab; our upper bound can just be $\mathcal{C}(a) \circ \mathcal{C}(b)$. But suppose we have a string of length 3: abb. Our upper bound could be $\mathcal{C}(a) \circ (\mathcal{C}(b) \circ \mathcal{C}(b))$ or $(\mathcal{C}(a) \circ \mathcal{C}(b)) \circ \mathcal{C}(b)$. We can now use the lattice structure to help: if $\mathcal{C}(abb) \leq X$ and $\mathcal{C}(abb) \leq Y$ then $\mathcal{C}(abb) \leq X \wedge Y$, since $X \wedge Y$ is a greatest lower bound. So in this case we can have our upper bound as $(\mathcal{C}(a) \circ (\mathcal{C}(b) \circ \mathcal{C}(b))) \wedge ((\mathcal{C}(a) \circ \mathcal{C}(b)) \circ \mathcal{C}(b))$. For longer strings, we have a problem: we cannot compute every possible binary tree and then take the meet over them all, since the number of such trees is exponential in the length.

However we can compute an even tighter bound using a recursive definition that can be computed by an efficient dynamic programming algorithm in $\mathcal{O}(|w|^3)$. For each substring of the word, we compute the lowest possible upper bound and then recursively combine them.

Given a language L and set of contexts F:

Definition 3. *we define* $\phi : \Sigma^* \to \mathfrak{B}(L, F)$ *recursively by*

- $\phi(\lambda) = \mathcal{C}(\lambda)$
- $\phi(a) = \mathcal{C}(a)$ *for all* $a \in \Sigma$, *(i.e. for all* $w, |w| = 1$*)*
- *for all* w *with* $|w| > 1$,

$$\phi(w) = \bigwedge_{u,v \in \Sigma^+ : uv = w} \phi(u) \circ \phi(v) \tag{4}$$

We can define the language generated by this representation to be:

$$\hat{L} = L(\mathfrak{B}(L, F)) = \{w | \phi(w) \leq \mathcal{C}((\lambda, \lambda))\} \tag{5}$$

We can show using a simple proof by induction that the computation ϕ will always produce an upper bound.

Lemma 1. *For any language* L *and set of contexts* F, *for any string* w, $\phi(w) \geq \mathcal{C}(w)$

This has the immediate consequence that:

Lemma 2. *For any language* L *and for any set of contexts* F, $L(\mathfrak{B}(L, F)) \subseteq L$.

As we increase the set of contexts, we will find that the language defined increase monotonically, until in the infinite limit when $F = \Sigma^* \times \Sigma^*$ we have that $L(\mathfrak{B}(L, \Sigma^* \times \Sigma^*)) = L$. This means that the problem of finding a suitable set of contexts is tractable and we can also define a natural class of languages

as those which have representations as finite lattices. We will call this class of representations the Distributional Lattice Grammars.

The class of languages definable by finite DLGS is denoted $\mathcal{L}_{\text{DLG}}$.

$$\mathcal{L}_{\text{DLG}} = \{L : \exists \text{ a finite set } F \subset \Sigma^* \times \Sigma^* \text{ such that } L(\mathfrak{B}(L, F)) = L\} \quad (6)$$

First we note that $\mathcal{L}_{\text{DLG}}$properly includes $\mathcal{L}_{\text{CCFG}}$. Indeed $\mathcal{L}_{\text{DLG}}$includes some non-context free languages, including ones that are related to the MIX language [22]. $\mathcal{L}_{\text{DLG}}$is a proper subclass of the languages defined by Conjunctive Grammars [23]. $\mathcal{L}_{\text{DLG}}$also includes a much larger set of context free languages than $\mathcal{L}_{\text{CCFG}}$includingsome non-deterministic and inherently ambiguous languages.

A problem is that the lattices can be exponentially large. We can however represent them lazily using a limited set of examples, and only compute the concepts in the lattice as they are needed. This allows for efficient learning algorithms. An important future direction for research is to exploit the algebraic structure of the lattice to find more compact representations for these lattices, using maximal elements.

7 Discussion

The importance of reducing the under-determination of the grammar given the language has been noted before: de la Higuera and Fernau [24] argue that learnable representations must have a canonical form, and that equivalence should be computable.

On a historical note, it is important not to neglect the contribution of the Kulagina school, which was initiated by the seminal paper of [25]. The first work on the distributional lattice was by Sestier [26], and subsequent work was developed by Kunze [27]. However the concatenation and residuation properties seem not to have been noted. Important papers that follow this line of research include [28, 29] as well as [30] and [31]. The ideas of basing representations on the congruence classes can be found in these works, but for some reason the rule schema $[uv] \to [u][v]$ seems not to have been discovered, instead exploration focussed on the linear grammar schema $[lur] \to l[u]r$, and on the development of contextual grammars [32]. We suspect that there are other related works in the Eastern European literature that we have not yet discovered.

Many other approaches can be recast in this form. For example the context-deterministic languages of [33], are exactly context free languages where the non-terminals have the property that they correspond to concepts, and where additionally the distributions of the non-terminals are disjoint: $\mathcal{C}(M) \vee \mathcal{C}(N) = \top$ for distinct non-terminals N and M.

There are a number of directions for future research that this approach suggests: probabilistic extensions of these algorithms, the development of algorithms for equivalence and decidability, and the use of these approaches for modelling transductions are all natural developments. Another extension would be to consider representations based on the relation $(x, y) \sim (u, v, w)$ iff $uxvyw \in L$. Some steps in this direction have recently been taken by Yoshinaka [34], which lead

naturally to Multiple Context-Free Grammars. There are numerous language theoretic problems that need to be explored in the use of these models.

Finally, representations such as these, which are both efficiently learnable and capable of representing mildly context-sensitive languages seem to be good candidates for models of human linguistic competence.

Acknowledgements

I am very grateful to Rémi Eyraud and Amaury Habrard.

References

1. Harrison, M.A.: Introduction to Formal Language Theory. Addison Wesley (1978)
2. Shieber, S.: Evidence against the context-freeness of natural language. Linguistics and Philosophy **8** (1985) 333–343
3. Chomsky, N.: The Minimalist Program. MIT Press (1995)
4. Chomsky, N.: Language and mind. 3rd edn. Cambridge Univ Pr (2006)
5. Harris, Z.: Distributional structure. In Fodor, J.A., Katz, J.J., eds.: The Structure of Language. Prentice-Hall (1954) 33–49
6. Angluin, D., Kharitonov, M.: When won't membership queries help? J. Comput. Syst. Sci. **50** (1995) 336–355
7. Evans, R., Gazdar, G.: DATR: A language for lexical knowledge representation. Computational Linguistics **22**(2) (1996) 167–216
8. Pollard, C., Sag, I.: Head Driven Phrase Structure Grammar. University of Chicago Press (1994)
9. Gold, E.M.: Complexity of automaton identification from given data. Information and Control **37**(3) (1978) 302 – 320
10. Carrasco, R.C., Oncina, J.: Learning deterministic regular grammars from stochastic samples in polynomial time. Theoretical Informatics and Applications **33**(1) (1999) 1–20
11. Clark, A., Thollard, F.: PAC-learnability of probabilistic deterministic finite state automata. Journal of Machine Learning Research **5** (May 2004) 473–497
12. Sénizergues, G.: The equivalence and inclusion problems for NTS languages. J. Comput. Syst. Sci. **31**(3) (1985) 303–331
13. Clark, A., Eyraud, R.: Polynomial identification in the limit of substitutable context-free languages. Journal of Machine Learning Research **8** (Aug 2007) 1725–1745
14. Clark, A.: PAC-learning unambiguous NTS languages. In: Proceedings of the 8th International Colloquium on Grammatical Inference (ICGI). (2006) 59–71
15. Clark, A., Eyraud, R., Habrard, A.: A polynomial algorithm for the inference of context free languages. In: Proceedings of International Colloquium on Grammatical Inference, Springer (September 2008) 29–42
16. Holzer, M., Konig, B.: On deterministic finite automata and syntactic monoid size. In: Proc. Developments in Language Theory 2002. (2002)
17. Angluin, D.: Inference of reversible languages. Communications of the ACM **29** (1982) 741–765
18. Clark, A.: A learnable representation for syntax using residuated lattices. In: Proceedings of the 14th Conference on Formal Grammar, Bordeaux, France (2009)

19. Conway, J.: Regular algebra and finite machines. Chapman and Hall, London (1971)
20. Lombardy, S., Sakarovitch, J.: The universal automaton. In Grädel, E., J., F., Wilke, T., eds.: Logic and Automata: History and Perspectives. Amsterdam Univ Pr (2008) 457–494
21. Lambek, J.: The mathematics of sentence structure. American Mathematical Monthly **65**(3) (1958) 154–170
22. Boullier, P.: Chinese Numbers, MIX, Scrambling, and Range Concatenation Grammars. Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL 99) (1999) 8–12
23. Okhotin, A.: Conjunctive grammars. Journal of Automata, Languages and Combinatorics **6**(4) (2001) 519–535
24. Fernau, H., de la Higuera, C.: Grammar induction: An invitation for formal language theorists. Grammars **7** (2004) 45–55
25. Kulagina, O.S.: One method of defining grammatical concepts on the basis of set theory. Problemy Kiberneticy **1** (1958) 203–214 (in Russian).
26. Sestier, A.: Contribution à une théorie ensembliste des classifications linguistiques. In: Premier Congrès de l'Association Française de Calcul, Grenoble (1960) 293–305
27. Kunze, J.: Versuch eines objektivierten Grammatikmodells I, II. Z. Zeitschriff Phonetik Sprachwiss. Kommunikat **20-21** (1967–1968)
28. Kříž, B.: Generalized grammatical categories in the sense of Kunze. Archivum Mathematicum **17**(3) (1981) 151–158
29. Drášil, M.: A grammatical inference for C-finite languages. Archivum Mathematicum **25**(2) (1989) 163–173
30. Novotny, M.: On some constructions of grammars for linear languages. International Journal of Computer Mathematics **17**(1) (1985) 65–77
31. Martinek, P.: On a Construction of Context-free Grammars. Fundamenta Informaticae **44**(3) (2000) 245–264
32. Păun, G.: Marcus contextual grammars. Kluwer Academic Pub (1997)
33. Shirakawa, H., Yokomori, T.: Polynomial-time MAT Learning of C-Deterministic Context-free Grammars. Transactions of the information processing society of Japan **34** (1993) 380–390
34. Yoshinaka, R.: Learning mildly context-sensitive languages with multidimensional substitutability from positive data. In Gavaldà, R., Lugosi, G., Zeugmann, T., Zilles, S., eds.: ALT. Volume 5809 of Lecture Notes in Computer Science., Springer (2009) 278–292