Grammar induction and language evolution

Jelle Zuidema and Rens Bod

ESSLLI 2008
20th European Summer School in Logic, Language and Information
4–15 August 2008
Freie und Hansestadt Hamburg, Germany

Programme Committee. Enrico Franconi (Bolzano, Italy), Petra Hendriks (Groningen, The Netherlands), Michael Kaminski (Haifa, Israel), Benedikt Löwe (Amsterdam, The Netherlands & Hamburg, Germany) Massimo Poesio (Colchester, United Kingdom), Philippe Schlenker (Los Angeles CA, United States of America), Khalil Sima'an (Amsterdam, The Netherlands), Rineke Verbrugge (**Chair**, Groningen, The Netherlands).

Organizing Committee. Stefan Bold (Bonn, Germany), Hannah König (Hamburg, Germany), Benedikt Löwe (**chair**, Amsterdam, The Netherlands & Hamburg, Germany), Sanchit Saraf (Kanpur, India), Sara Uckelman (Amsterdam, The Netherlands), Hans van Ditmarsch (**chair**, Otago, New Zealand & Toulouse, France), Peter van Ormondt (Amsterdam, The Netherlands).

`http://www.illc.uva.nl/ESSLLI2008/`
`esslli2008@science.uva.nl`

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

ESSLLI 2008 is organized by the Universität Hamburg under the auspices of the *Association for Logic, Language and Information* (FoLLI). The *Institute for Logic, Language and Computation* (ILLC) of the *Universiteit van Amsterdam* is providing important infrastructural support. Within the Universität Hamburg, ESSLLI 2008 is sponsored by the Departments *Informatik*, *Mathematik*, *Philosophie*, and *Sprache, Literatur, Medien I*, the *Fakultät für Mathematik, Informatik und Naturwissenschaften*, the *Zentrum für Sprachwissenschaft*, and the *Regionales Rechenzentrum*. ESSLLI 2008 is an event of the *Jahr der Mathematik 2008*. Further sponsors include the *Deutsche Forschungsgemeinschaft* (DFG), the Marie Curie Research Training Site GLoRiClass, the European Chapter of the Association for Computational Linguistics, the *Hamburgische Wissenschaftliche Stiftung*, the Kurt Gödel Society, Sun Microsystems, the Association for Symbolic Logic (ASL), and the European Association for Theoretical Computer Science (EATCS). The official airline of ESSLLI 2008 is Lufthansa; the book prize of the student session is sponsored by *Springer Verlag*.

Jelle Zuidema and Rens Bod

Grammar induction and language evolution

Course Material. 20th European Summer School in Logic, Language and Information (ESSLLI 2008), Freie und Hansestadt Hamburg, Germany, 4–15 August 2008

Grammar Induction & Language Evolution

Willem Zuidema
Rens Bod

ESSLLI 2008, Hamburg

Course description

Recently, much progress has been made in developing methods for learning grammars from natural language text. The first part of the course gives an overview of such algorithms, starting with those which move through a space of context-free grammars using nonterminal chunking, merging and/or splitting. We discuss several local search heuristics, as well as some global objective functions (Bayesian posterior probability, maximum likelihood) and inference procedures (e.g. EM). We evaluate strengths and weaknesses of these models on several metrics, and consider some extensions that work with richer input data (phonological phrasing, semantics) and different grammatical formalisms (dependency grammar, categorial grammar, TAG, DOP). In the second part of the course, we discuss applications of these algorithms in models of language acquisition, change and evolution. We show how the constraints on the search space necessary for successful learning emerge automatically in iterated learning. We consider biological evolution of the inductive bias, and look at the nativist-empiricist controversy from this perspective.

Lecturers

Willem Zuidema, *email:* jzuidema@science.uva.nl; *address:* Institute for Logic, Language and Computation, University of Amsterdam, Plantage Muidergracht 24, 1018 TV, Amsterdam, The Netherlands

Rens Bod, *email:* rens@science.uva.nl; *address:* Institute for Logic, Language and Computation, University of Amsterdam, Plantage Muidergracht 24, 1018 TV, Amsterdam, The Netherlands

Course materials

Included are the following papers:

- Yoav Seginer (2007), *Learning Syntactic Structure*, PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam; **Chapter 1**: Introduction
- Rens Bod, *Introduction to Elementary Probability Theory and Formal Stochastic Language Theory*, unpublished.
- Gideon Borensztajn and Willem Zuidema (2007), *Bayesian Model Merging for Unsupervised Constituent Labeling and Grammar Induction.* ILLC Prepublication PP-2007-40, Institute for Logic, Language and Computation, University of Amsterdam.
- Rens Bod (2006), *An All-Subtrees Approach to Unsupervised Parsing.* Proceedings ACL-COLING 2006, Sydney.
- Rens Bod, 2007, *Is the End of Supervised Parsing in Sight?* Proceedings ACL 2007, Prague, 400-407.
- Rens Bod (2008), *From Exemplar to Grammar: Integrating Analogy and Probability in Language Learning*, ILLC Prepublication PP-2008-23, Institute for Logic, Language and Computation, University of Amsterdam.
- Willem Zuidema (2003), *Modeling language acquisition, change and variation*, Proc. Workshop on Language Evolution and Computation, ESSLLI'03, Vienna.
- Willem Zuidema (2002), *Language adaptation helps language acquisition*, Proc. International Workshop on Self-Organization and Evolution of Social Behaviour, Monte Verita, Switzerland
- Willem Zuidema (2005), *The major transitions in the evolution of language*, PhD thesis, Theoretical and Applied Linguistics, University of Edinburgh; **Chapter 2**: The evolutionary biology of language
- Willem Zuidema (2008), *An annotated bibliography of grammar induction models for natural language learning*, unpublished.

Learning Syntactic Structure

Yoav Seginer (2007)

PhD thesis
Institute for Logic, Language and Computation
University of Amsterdam

Chapter 1

Introduction

Syntactic structure plays a central role in most theories of language, but it cannot be directly observed. An important question, therefore, is whether there is a relation between syntactic structure and immediately observable properties of language, such as the statistics of the words and sentences that we hear and read. Finding such a relation has important consequences for the problem of language acquisition by children, as well as implications for the theory of syntax itself. It can also be used in engineering language processing systems.

This thesis addresses the problem of finding the relation between the surface statistics of a language and its hidden syntactic structure by developing a parser which attempts to capture this relation. The parser is tested to determine its agreement with the syntactic structure which linguists assign to utterances. While this approach does not try to model the way humans learn and process language, the design of the parser relies on some well-known properties of language and language processing by humans. By selecting both the representation of syntactic structure and the language statistics in a way which agrees well with these properties of language, the resulting relation, as coded by the parser, is simple.

1.1 Background

One of the more notable facts about language is that children can learn it without explicitly being informed of its structure and meaning, as already observed by Saint Augustine in his *Confessions* (I 8):

> Passing hence from infancy, I came to boyhood, or rather it came to me, displacing infancy. Nor did that depart (for whither went it?) and yet it was no more. For I was no longer a speechless infant, but a speaking boy. This I remember; and have since observed how I learned to speak. It was not that my elders taught me words (as,

> soon after, other learning) in any set method; but I, longing by cries and broken accents and various motions of my limbs to express my thoughts, that so I might have my will, and yet unable to express all I willed, or to whom I willed, did myself, by the understanding which Thou, my God, gavest me, practise the sounds in my memory. When they named any thing, and as they spoke turned towards it, I saw and remembered that they called what they would point out by the name they uttered. And that they meant this thing and no other was plain from the motion of their body, the natural language, as it were, of all nations, expressed by the countenance, glances of the eye, gestures of the limbs, and tones of the voice, indicating the affections of the mind, as it pursues, possesses, rejects, or shuns. And thus by constantly hearing words, as they occurred in various sentences, I collected gradually for what they stood.[1]

Saint Augustine's theory of language learning certainly may sound plausible to the modern reader,[2] but, whether correct or not, it merely describes the learning of the meaning of words (and only of some words, for that matter). This is at best a modest beginning, since to properly understand and use language, a child does not only need to learn the meaning of individual words, but must also understand how these can be combined to produce complex linguistic structures, such as sentences. The way the words are arranged together is described by the syntax of the language, and different languages not only use different words, but also have different syntactic structures. A child must, therefore, not only learn the meaning of individual words but must also learn the rules governing the syntax of the language. Because these rules are abstract, they cannot be learned simply by associating them with objects in the real world. At the same time, this abstract nature of syntactic rules and their relatively loose connection with meaning have made syntax one of the most formally described components of natural language. For these formal syntactic systems, concrete learning algorithms can be designed and tested. Such an algorithm is the subject of the present work.

When studying natural language learning, one may consider various methods by which the rules governing the use and interpretation of a language can be deduced from exposure to utterances of that language. There may be different ways and settings in which this goal can be achieved and the way children acquire their first language may represent only one possible method for doing so. Here I will use *language acquisition* to refer to *child language acquisition*, the specific process by which children learn their first language, while *language learning* will refer more generally to any method of learning.

Language acquisition by children is, of course, not merely a specific instance of natural language learning but also the most successful learning method known

[1] English translation by Edward Bouverie Pusey.

[2] who has not read Wittgenstein's *Philosophical Investigations*.

to date. This is not surprising, as it is only the successful learning of language by children which allows natural languages to exist at all and the acquisition process thus defines the range of possible languages (Deacon 1997; Kirby and Hurford 2002; Zuidema 2003). The possibility remains, however, that additional learning methods exist. This is especially true if the setting in which learning takes place is not identical to that in which children learn their first language. For example, a computer may be required to learn the syntax of a language from written text. This input is clearly very different from the speech signal which a child is exposed to when acquiring a language, and may require different learning algorithms. This does not mean, however, that such learning algorithms are not relevant to the study of child language acquisition. I will discuss this later on in the introduction.

There are many ways to approach the problem of language learning and of syntax learning in particular. These approaches differ not only in the solutions they offer, but in the questions they pose and in the way they set out to answer these questions. Of the many roads available, I have chosen to travel down one: to design and test empirically (on large corpora of text) an algorithm which learns to parse from unannotated example sentences. This thesis does not, therefore, address the question of child language acquisition directly but is concerned with the more general problem of language learning.

Before going down this road, the introduction is a fitting place to take a quick look down the roads not taken. I therefore begin with a very brief mention of other approaches to the study of language learning and acquisition: in psychology, in theoretical linguistics and in theoretical computer science. This should allow the reader to place the current work within a wider context, but the emphasis is on mentioning, rather than discussing, the main alternative approaches. I then go on to explain what one may expect to learn from the approach adopted in this thesis, which uses real language input (though not necessarily that which is available to children) to simulate learning on a computer (rather than observing it in children). Next, taking a first step down the road chosen, I look more closely at previous computer algorithms designed to learn syntactic structure in settings similar to those I use. Having thus looked at all the roads I could have taken, I conclude the introduction by describing the road I have taken and those properties of it which I find most attractive.

1.2 Some Roads not Taken

1.2.1 Psychological and Linguistic Approaches

The obvious way to study language acquisition is to observe small children as they learn their first language. This has become a thriving field of research within modern psychology (see Ingram (1989) for work up to the late 80's and Tomasello

and Bates (2001) for more recent work). This research has produced an impressive body of observations and theory, but it remained largely incompatible with most of the theories linguists developed for adult language. This is no major problem when studying the first stages of language acquisition in very young children, but as older children, acquiring more complex linguistic skills, become the subject of research, the problem becomes increasingly pressing. From the point of view of many linguists, the psychological study of the development of language in children has failed to deal with the true complexities of adult language which must be acquired by children (Pinker 1984).

An alternative approach to the study of language acquisition has developed within the field of linguistics, with the end point of the acquisition process, the adult language, as its starting point. Following Chomsky (1965), linguists compare different languages to identify those properties which all languages have in common. This approach has become known as *principles and parameters* (Chomsky and Lasnik 1993). The assumption is that a learning procedure only has to be defined for the idiosyncratic properties of each language while the common properties of all languages can be assumed to be innate. This has been the main program of Chomskian linguistics, which attempts to identify a *universal grammar* for all languages and a set of parameters which distinguishes different languages and must be learned.[3] Using this framework, one could then hope to be able to go back to the child development data and discover the exact way in which children actually discover the values of the parameters for the specific language they are exposed to.

Chomsky (1965) distinguished between linguistic theories which have *descriptive adequacy* and theories which also have *explanatory adequacy*. A theory with descriptive adequacy correctly describes the structures found in different languages while a theory which has explanatory adequacy must also explain how the rules used to describe the structures of each language can be deduced from examples of that language. Chomsky (p. 26) claims that "gross coverage of a large mass of data can often be attained by conflicting theories; for precisely this reason it is not, in itself, an achievement of any particular theoretical interest or importance." Therefore, only the condition of explanatory adequacy can allow us to decide between the conflicting theories. In this way, the problem of language acquisition (and specifically, the acquisition of syntax) has moved center stage in linguistic study. Even without actually achieving explanatory adequacy (something which has not yet been achieved), the need for it has been a driving force behind linguistic research, especially within the Chomskian tradition.

One could imagine the child and adult centered approaches to the study of language acquisition working towards each other, but instead, two belligerent camps have formed, with child centered approaches attempting to stretch child language

[3]Chomsky (1965) points out that the idea of a universal grammar goes at least as far back as the 18th century.

all the way up to adult language and adult centered approaches attempting to stretch adult language all the way down to child language. Comprehensive theories covering the full process of language acquisition have been developed within both approaches (see Tomasello (2003) for an example of the child centered approach and Pinker (1984) for an acquisition theory based on formal mechanisms developed for adult language). The two approaches remain largely incompatible, probably because they seem to align well with opposing stances in traditional controversies about language and human cognition in general, such as the debate about nativism vs. empiricism. Child centered approaches to language acquisition, the modern variant of which are often *usage based theories* (Langacker 1987; Langacker 1991; Tomasello 2006), tend to take an empiricist stance and emphasize the use of general cognitive capabilities of abstraction, generalization and analogy in language acquisition and use. In contrast, adult centered theories are based on some form or other of universal grammar, which is often complex and is assumed to be an innate, language specific, human capability, thus taking a nativist stance. This debate is still raging, but as it is mainly a debate about language acquisition and not about language learning, it is of no concern to the present work.

1.2.2 Theoretical Mathematical Models

Because much of what happens in the process of learning a language is invisible to us, researchers have been concerned with identifying settings which allow languages to be learned in principle (whether children actually use those methods or not). The hope is that the range of theoretical possibilities (once identified) is sufficiently constrained to allow for the correct theory to be selected based on the observed behavior of children. Linguists have been pursuing such a goal in their search for universal grammar by attempting to identify the similarities between different existing languages. As syntactic theory became increasingly formalized in twentieth century linguistics, it became possible (and tempting) to try to formalize the process by which the syntax of a particular language can be learned from example sentences of that language. Researchers in the field of *computational learning theory* who are interested in *grammatical inference* look for mathematical models which would allow them to define learnable classes of languages and algorithms for selecting one language out of such a class based on example sentences from the target language.

The seminal theoretical work in this field is Gold (1967), which defined an abstract model of the learning process and criteria for successful learning. The learner is assumed to receive one example after the other from the target language and can, at each step, make a guess as to the grammar of that language. The learner is allowed to err at first, but after a finite number of steps must converge to a correct grammar. Gold showed in his paper that if the learner assumes that the grammar can be any context free grammar and if all the learner has to go on

is an arbitrary sequence of sentences in the language, then there is no algorithm which guarantees convergence to a correct grammar. In Gold's terminology, this means that the class of context free grammars is not learnable in the limit from positive examples. Because most linguists believe that a grammar has to be at least context free to allow for the phenomena observed in natural language syntax, Gold's negative result has been widely (and sometimes wildly) cited in the linguistic and cognitive literature (see Johnson (2004) for discussion). It was used, among others, to support the innateness of language or to dismiss Gold's paradigm of learning altogether (since, as we know, languages are learnable).

While Gold's theorem was for many a final statement (for good or bad), for many others (including Gold) it was only a starting point. In the decades that followed, researchers looked for variations of the original setting which could allow linguistically relevant classes of languages to be learnable. One possible variation is to change the definition of success, for example by introducing a probabilistic success criterion (e.g. Valiant 1984). Another approach is to search for specific learnable classes of grammars within the original setting of Gold's paper (e.g. Kanazawa 1998). Gold's negative results apply to large classes of grammars, but they do not necessarily hold for sub-classes of these classes. Because many languages in the classes Gold used are not plausible candidates for human languages, sub-classes of these classes, which better describe the range of possible human languages, may be learnable. Several surveys of these results are available (Lee 1996; Sakakibara 1997; de la Higuera 2005). While this line of research has created a significant body of theoretical results, its relevance to the empirical study of language acquisition remains limited because the grammatical systems considered were usually not powerful enough by the standards of modern theoretical and computational linguistics and because even when positive learnability results were achieved, they often made unrealistic assumptions about the input and resources available to the learner (such as noiseless input or very long convergence time). While some algorithms were implemented, most were not tested on real natural language data.

1.3 The Road Taken: Empirical Computational Models

Even before Gold's theoretical work and possibly also before Chomsky's introduction of explanatory adequacy as a goal for theoretical linguistics, researchers in artificial intelligence were attempting to build computer systems which could learn grammars from text (Lamb 1961). These efforts continued from the early 60's until today, ranging from simulations (working with toy grammars and languages) to systems applied to large corpora of real language. The algorithm described in the present work falls into this last class of models and has been applied to large collections of natural language sentences (see chapter 7).

While some of these computer systems (and especially the simulations) were designed to simulate the process of language acquisition by children, an important difference between this line of research and previously mentioned approaches to the study of language learning is that often the immediate motivation behind the construction of language learning computer systems is not necessarily to shed light on the problem of language acquisition by children but, instead, the need to solve some engineering problem. This is not to say that researchers developing such algorithms ignore the question of language acquisition by children, but it does mean that they do not feel committed to psychologically plausible algorithms and are driven more by the success of the algorithm on the specified task rather than its psychological modeling accuracy.

1.3.1 The Task

When it comes to syntax, the main (but not only) engineering task studied is parsing. A parser is an algorithm which takes an utterance as input and outputs the syntactic structure of that utterance. Since the syntactic structure of an utterance cannot be directly observed, different linguistic theories may assign different syntactic structures to the same utterance. The syntactic structures defined by some of these theories may be very complex, but at the most basic level syntactic structure is either described in terms of dependency links (from one word to another) or by grouping words together into syntactic units. Dependency links indicate that some relation holds between the words (such as the relation between an adjective and the noun it modifies or between a verb and its object). The grouping of words into units reflects the observation that these groups of words can function as a unit, or a *constituent*. For example, in the sentence *the dog barked*, the two words *the dog* can be replaced by a single pronoun *it*. This implies that in some ways *the dog* is a single unit. While the most obvious examples of dependencies and constituents are non-controversial, there are many cases which are debatable. Therefore, the construction of a parser always implies a choice of syntactic theory. When working with annotated corpora, one often adopts the decisions made by the annotators as to the syntactic structure. I will do so as well, but I will also discuss some of the choices made by the annotators in chapter 4.

The learning algorithms I study here are algorithms which learn to parse a language by examining unannotated example sentences. This can be referred to as *unsupervised parsing*, in contrast to *supervised parsing*, where an algorithm learns to parse a language from syntactically annotated examples.

1.3.2 Why Study Unsupervised Parsing?

Both supervised and unsupervised parsers use some form of learning to replace the traditional method of designing parsers by manually writing a set of gram-

mar rules for each language. While using a supervised parser reduces the effort involved in writing grammar rules, it also requires a significant amount of manual labor because for each new language (and domain) to be parsed, one needs to syntactically annotate a large enough corpus of text. In an unsupervised learning approach, however, all that one needs to do is feed the learning algorithm with sufficiently many unannotated examples of the target language and domain. Since large amounts of electronic text are now available for many languages, this approach is by far the cheapest method. It is therefore appealing, from an engineering point of view, to have algorithms which can learn to parse in an unsupervised way. While such algorithms are yet to achieve parsing accuracy even close to that achieved by other methods, recent years have seen a new surge of interest in the development of unsupervised parsing algorithms, with significant improvement over past results (Klein and Manning 2002; Klein and Manning 2004; Dennis 2005; Bod 2006a; Bod 2006b).

Beyond the engineering motivation, the implementation of learning computer systems remains highly relevant to the study of language learning because it is the only approach which can test what happens when a proposed algorithm interacts with real language input. While psychological and linguistic theories are mostly based on individual examples picked out by the researchers and while mathematical models only make general theoretical assumptions about the input available to the learner, implementing computer models and testing them on large corpora of real language allows the cumulative effect of numerous examples to be studied. In this respect, this approach is the closest to studying language acquisition in its natural settings.

Even when a computational model is clearly not psychologically realistic, its success in learning syntactic structure has important implications to the study of language and language acquisition because such successful learning indicates a relation between the surface structure of a language and its hidden syntactic structure. Even if the method by which this relation is established is not actually used by children acquiring a language, the relation is still an empirical property of the language and may be used by children in some other way in the process of language acquisition.

This brings us to another reason for developing learning algorithms for syntactic structure, one which goes back to Harris (1946). In that paper, Zellig Harris proposed a "formalized procedure for describing utterances directly in terms of sequences of morphemes" which covers "an important part of what is usually included under syntax." The procedure proposed by Harris, which groups together sequences of words by the contexts in which they appear, is known as the *distributional* method and has become the starting point of many modern grammar induction algorithms. Harris himself, however, was not interested in the problem of language acquisition but rather in providing an explicit procedure for describing syntactic structure to replace "the use of diverse undefined terms and a reliance on semantic rather than formal differentiation" in the description

of syntactic structure. Harris thus attempted to establish syntactic analysis as an empirical science which has sequences of words uttered by a speaker rather than the cognitive processes taking place in that speaker's mind as its subject matter. This approach, which is behaviorist in nature, lost much of its popularity (together with behaviorism in general) after the cognitive revolution of the 50's. Beyond the general shift in linguistics from the study of surface structure to the study of the cognitive processes involved in language processing, one of the reasons that linguists abandoned distributional methods in the study of grammar is their failure to achieve significant results, just as grammar induction algorithms failed for many years to achieve even modestly good results on real language input. Whether a purely distributional approach can indeed teach us anything important about the syntactic structure of language remains to be seen, but I believe that if successful algorithms can be designed to infer syntactic structure from unannotated examples then this should certainly have implications for the theory of syntax and can serve as an empirical method (which does not rely on human judgments) for discovering the syntactic structure of language. Just as one does not need to be a behaviorist to study behavior, one does not need to deny the relevance of cognition to linguistics in order to use distributional methods in the study of language. While current methods are still too weak to contribute directly to the study of syntax in the way Harris envisioned it, advances made in recent years may indicate that algorithms can discover at least some of the syntactic structure of a language. In this way, Harris's original program of coming up with a formalized procedure for describing an "important part of what is usually included under syntax" remains a valid goal for research with significant benefits to our understanding of language; if it succeeds.

1.3.3 On the Use of Meaning in Learning Syntax

Whatever the approach taken, acquisition of language by children remains relevant because the fact that children can learn language without getting explicit information about its structure means that language learning is possible, at least in principle. Of course, there is significant debate as to the exact information available to children when they acquire their language (and specifically syntax) and it is clear that a computer can never be exposed to the full experiences of the child. One central question is whether syntax can be learned independently from meaning (semantic or pragmatic). Many theories of language acquisition, such as Pinker (1984), are based on *semantic bootstrapping*, where the child first learns the meaning of some words and then uses this acquired knowledge to deduce the syntactic structure of sentences in which these words appear. While this seems to be a simple process for acquiring syntactic knowledge, it has also been shown (Landau and Gleitman 1985; Gleitman 1990) that knowledge of syntactic structure is necessary for the correct learning of the meaning of many words, when a linguistic utterance can only be mapped onto an observed situation if the

syntactic structure of the utterance is known to the child. This process of *syntactic bootstrapping* can work together with semantic bootstrapping to learn both meaning and syntax. How the two may be combined and which role is played by each component remains an open question.

Implementing semantic bootstrapping in a fully formalized system has been attempted by several researchers, both theoretically, within Gold's paradigm of learning (Hamburger and Wexler 1975; Tellier 1998; Dudau-Sofronie et al. 2001; Oates et al. 2003), and in actual computer systems (Anderson 1977). The main problem with all these systems is that they do not learn the semantics as a child does, but take the semantic representation as input together with the utterance describing the situation. These semantic representations are stipulated by the designers of the algorithms, and can all be suspected of encoding syntactic information which needs to be learned by the child. The discussion whether these properties are semantic or syntactic is irrelevant, as the question remains how a child can learn them from the input available. Moving the syntax into the semantics does not solve the problem, but only avoids it. This is also the reason why computer systems designed to use semantics (such as Anderson's) were toy systems designed to prove a cognitive theory rather than systems designed towards a specific application. From an engineering point of view, to use the semantic information required by these systems would require semantically annotated corpora, so for all practical purposes it is simpler to use syntactic annotations to begin with.

Because the most readily available input for a computer program attempting to learn the syntax of a language is unannotated sentences (without any information about their meaning or context) most algorithms remain entirely distributional in nature. The objection that this is not the way children learn their language does not detract from the importance of these algorithms, if they are successful. Success of such algorithms is both useful in constructing language processing systems and in understanding the relations between the surface structure of language and its syntactic structure. This is also the approach I adopt in the present work.

1.3.4 A Brief Survey of Syntactic Induction

Over the years, many systems for learning the syntax of natural languages were proposed and implemented. This section is a brief survey of these systems and the principles used in their design. I discuss only systems which were actually implemented and which take unannotated example utterances of a language as input.

Until recently, most of these systems learned a *context free grammar*. A context free grammar (CFG) consists of a finite set of rules of the form $X \rightarrow Y_1, \ldots, Y_n$, which can be used to replace (rewrite) the symbol X by a sequence of symbols $Y_1, \ldots, Y_n$. Beginning with a single *start symbol* S, rules can be applied

repeatedly until a sequence of words is formed (X cannot be a word, so words cannot be rewritten). Given a sentence to be parsed, a CFG parser looks for a sequence of rule applications which generates that sentence from S. The sequence of rule applications defines the syntactic units (constituents) of the sentence: the sequence of words which was generated from a single symbol X is a unit and X is the label of that unit (e.g. X = noun phrase). Since there may be more than one way to generate a sentence with the same CFG, the parser must have a way to select one of the possible parses. A standard way of doing so is to use a *probabilistic context free grammar* (PCFG) in which every rule is assigned a probability (the probabilities of all rules with the same left hand side must sum to 1). The rule probabilities induce a probability for each parse and the parser selects the most probable one.[4]

Most of this section is dedicated to the description of various algorithms which learn the syntax of a language by inducing a (probabilistic) context free grammar. At the end of this section I describe some more recent algorithms which do not use CFG induction but instead define various ways of inducing a parser directly. These algorithms turn out to be far more successful than the older CFG induction algorithms and I conclude the section with a short discussion of what is, in my opinion, the main reason for this difference.

Distributional Clustering

Many grammar induction algorithms use a method of *distributional clustering* which may be traced back to Harris (1946):

> The procedure [...] consists essentially of repeated substitution: e.g. *child* for *young boy* in *Where did the — go?*. To generalize this, we take a form A in an environment C —— D and then substitute another form B in place of A. If, after such substitution, we still have an expression which occurs in the language concerned, i.e. if not only CAD but also CBD occurs, we say that A and B are members of the same substitution-class, or that both A and B fill the position C —— D , or the like.

Because Harris was not interested in language induction but in structural description, he allowed the decision as to whether CAD and CBD are in the language to be taken by a linguist. When this procedure is used to induce a grammar, the decision whether A and B are members of the same substitution class is based not on the linguistic judgments of a linguist but on the occurrence of both CAD and CBD in a corpus of utterances in the language being learned. This method has been the cornerstone of many grammar induction algorithms

[4]For more detailed definitions of CFG and PCFG, see, for example, Jurafsky and Martin (2000).

beginning with Lamb (1961), and has since been used also by many others (Cook et al. 1976; Wolff 1982; Mori and Nagao 1995; Adriaans et al. 2000; van Zaanen 2000; Clark 2001; Solan et al. 2005). Some additional algorithms from the 60's and 70's based on this method are mentioned in the survey of Pinker (1979). The term *environment* used by Harris has been replaced by *context* in the more recent literature.

While the idea seems simple and straightforward, there are several fundamental problems in implementing it successfully. Several of these were already mentioned in Harris's original paper. The first problem that Harris mentions is that:

> In some languages, relatively few morphemes occur in exactly the same environments as others: *poem* occurs in *I'm writing a whole — this time* but *house* does not. Both morphemes, however, occur in *That's a beautiful —*. Shall we say that *poem* and *house* belong in general to the same substitution class, or that they have some environments in common and some not?

This problem worsens when it is not a linguist who has to decide whether a certain sentence appears in a language but a corpus is used to make such decisions. Even a large corpus contains only a small fraction of the utterances which may reasonably be produced in a language and even if two sequences of words can, in principle, appear in a certain environment, it may very well be that no evidence for this will be found in the corpus. To solve this problem, algorithms generally do not require that sequences of words appear in exactly the same contexts in order to cluster them together and some overlap of contexts is considered sufficient. The exact criterion used may vary from simple clustering of any two sequences appearing in the same context (van Zaanen 2000) to complex algorithms based on the combination of different contexts (Adriaans et al. 2000).

Part-of-Speech Induction

One task on which the clustering by context technique has proven successful is the induction of parts-of-speech, that is, the assignment of a class label to each word. This is a subtask of the general clustering task because it only considers single words (rather than sequences of words) for substitution. Using only the most frequent words in the corpus as contexts, various clustering methods have been used to induce part-of-speech tags (Schütze 1995; Clark 2000). This is not only a useful result in itself but may also serve as a first step in the induction of a grammar. For this reason, most recent syntactic induction algorithms take sequences of part-of-speech tags rather than words as their input. This considerably simplifies the problem of identifying sequences appearing in identical contexts, because both the sequences and the contexts are drawn from a much

smaller set of possible symbols. In practice, the part-of-speech tag sequences are often taken from an annotated corpus rather than being induced.

Identifying Constituents

A second problem with the distributional method identified by Harris is that when *sequences* of words are considered for substitution, the substitution classes created by the method may contain sequences of words which are not constituents at all:

> Since our procedure now permits us to make any substitution of any sequences, it may become too general to produce useful results. For example, we might take the utterance *I know John was in* and substitute *certainly* for *know John*, obtaining *I certainly was in.* This substitution conceals the fact that the morphemes of *I know John was in* can be said as two utterances instead of as one.

Harris goes on to mention other respects in which *certainly* and *know John* differ and then suggests that "substitution of sequences be so carried out as to satisfy all manipulations of that environment which forms the frame of the substitution." Even if such a procedure can be carried out by a linguist, it certainly cannot be carried out by an algorithm which only has a small subset of the utterances in the language to work with and does not know how to identify all permissible manipulations of a given environment.

For this reason, some clustering-based induction algorithms (Mori and Nagao 1995; Clark 2001) explicitly define a procedure to distinguish between sequences (of part-of-speech tags) which are constituents and those which are not. Mori and Nagao make the assumption that sequences (of part-of-speech tags) which represent constituents are less constrained as to what precedes and follows them than non-constituent sequences and implement this by setting a threshold on appropriate conditional entropy functions for the right and left contexts of a sequence. Clark uses a criterion which is based on the assumption that the mutual information between the left and right context of a sequence is higher for constituents than for non-constituents. These two methods seem to implement similar intuitions in different ways. The algorithm of Solan et al. (2005) does not explicitly distinguish between constituents and non-constituents but does seem to use a method similar in nature (but not in detail) to that of Mori and Nagao (1995) in order to detect "significant patterns" which eventually become the constituents of the analysis. Other clustering algorithms do not have an explicit procedure for identifying constituents, but instead rely on the grammar rule construction procedure (described below) to implicitly prefer rules describing constituents.

Inducing the Grammar Rules

Having clustered sequences of symbols and possibly having determined which of these are candidate constituents, all sequences in a cluster can be replaced, wherever they appear in the corpus, by a new single symbol representing the cluster. This is represented by defining a set of context free rules which have the new symbol as their left hand side and the sequences in the cluster as their right hand side. Because the sequences in different clusters may overlap, replacing all occurrences in the corpus of sequences from one cluster by a single symbol can destroy the sequences which are part of another cluster. For this reason, the induction algorithm must determine which cluster to substitute first. Having performed the substitution, the process can be repeated.

Most algorithms (Cook et al. 1976; Wolff 1982; Mori and Nagao 1995; Clark 2001) use an objective function to decide which grammar rule to create at each step. These objective functions are all similar in nature (but not necessarily in detail) and may be traced back to Solomonoff (1964), who defined a Bayesian probability function which has to be maximized by the grammar induction algorithm. This probability function is $P(D|G)P(G)$, where $P(G)$ is the a-priori probability of the grammar and $P(D|G)$ is the probability of the observed data (corpus) given the grammar. The a-priori distribution is usually taken to be such that smaller grammars have higher probability. Maximizing the Bayesian probability function is equivalent to minimizing $-\log(P(D|G)) - \log(P(G))$ which is a description length criterion. The quantity $-\log(P(G))$ is seen as describing the size of the grammar and $-\log(P(D|G))$ is seen as the length of the data after being encoded by the grammar. This is often interpreted as a compression criterion because a good grammar which captures the regularities of a language should allow the data to be encoded compactly. While details vary, most algorithms use some variant of this function (either in its Bayesian or description length form) for rule selection. An exception is Solan et al. (2005), who uses the "most significant pattern" (which resembles the constituency criterion of Mori and Nagao 1995) as a criterion for substitution.

Because substituting a single symbol for a constituent immediately destroys all non-constituent sequences which overlap (but do not contain) that constituent, the process of rule selection can potentially eliminate non-constituent clusters. The burden of doing so correctly is placed on the objective function by which rule selection is determined. By filtering out non-constituent clusters before the rule selection step, Mori and Nagao (1995) and Clark (2001) increase the chances of this happening.

The algorithms of van Zaanen (2000) and Adriaans et al. (2000) are an exception to this process in that they do not substitute and re-cluster after each substitution but instead continue to use the clustering on the original text. While van Zaanen (2000) proposes different heuristics to decide between conflicting constituents in the text, Adriaans et al. (2000) simply create a set of rules from their

clustering without going back to the original text (and thus do not have to deal with conflicting constituents). Of course, when parsing with these rules, only non-crossing units are created, but it is not entirely clear whether there is any mechanism in the algorithm which allows constituent clusters to be preferred over non-constituent clusters.

Syntagmatic and Paradigmatic Merging

When a clustering algorithm creates a grammar rule and substitutes all sequences in a cluster by the left hand side (non-terminal) symbol of the rule, it actually makes two decisions: first, it identifies each of the sequences as a constituent and, second, it identifies these constituents as being substitutable for each other. Borrowing structuralist terminology, some authors (Wolff 1982; Stolcke 1994) refer to these operations as syntagmatic merging (grouping words into syntactic units) and paradigmatic merging (grouping units into substitution classes). Stolcke (1994) also names them *chunking* and *merging* (respectively). In a chunk (syntagmatic merge) step, a single sequence of symbols is replaced, wherever it appears in the corpus, by a new non-terminal symbol and an appropriate context free rule is added to the grammar. In a merge (paradigmatic merge) step, several different non-terminals are merged into a single non-terminal. This approach was already applied in the algorithms of Cook et al. (1976) and Wolff (1982). Because chunking is used, one can restrict merging (clustering) to single symbols appearing in the same context, rather than having to cluster sequences of different lengths as in the original Harris method. Cook et al. make use of this and only allow merging of single symbols while Wolff seems to retain the possibility of merging sequences of symbols (in addition to chunking). Both algorithms decide which of the many possible chunking or merging operations to perform at each step based on the improvement on an objective function resulting from such an operation.

Stolcke (1994) goes one step further and does not use context at all as a criterion for merging. Instead, merging can be performed between any two non-terminals and which merge or chunk to perform depends only on the improvement on an objective function resulting from such a merge. Thus, the full burden of success is put on the shoulders of the objective function and its correct design becomes critical. It is interesting to note that Stolcke (p. 88) observes that chunking often must be combined with merging to achieve an improvement on the objective function. This seems to suggest that while the separation of merging and chunking is conceptually elegant, the two operations must be performed together and the separation is in practice undone.

Highest Likelihood PCFG

The induction algorithms mentioned so far are sometimes referred to as *structure search* algorithms, because they search for the grammar which optimizes the objective function by constructing a set of grammar rules. An alternative is *parameter search*, in which the set of possible rules is fixed and only the probability of each rule (in a probabilistic context free grammar) has to be determined. The search is then for the probability distribution which maximizes the likelihood of the observed data. This process may assign some rules a zero (or very small) probability, thus eliminating them effectively from the grammar. It is therefore possible to start with a relatively large set of possible rules and hope that the parameter search will only assign large probabilities to a small subset of them.

Of the two components of the objective function used in the structural search algorithms, we are left with only one: $P(D|G)$, the likelihood of the data given the probabilistic grammar. The a-priori probability of the grammar is no longer used. This may seem to suggest that the a-priori probability of the grammar is not needed to begin with, but this is not true. Parameter search algorithms must restrict the possible grammar rules they allow because, otherwise, the maximum likelihood is achieved by the trivial grammar in which every sentence in the corpus is generated by a single rule and the probability of the rule is equal to the relative frequency of the sentence in the corpus. The selection of the initial set of grammar rules to which a non-zero probability may be assigned becomes a critical issue in the design of parameter search algorithms. While this may be difficult to achieve in a way which is not biased towards specific languages, it is also probably not reasonable to assume that all context free grammars should remain a-priori possible, as is assumed by most structure search algorithms.

One advantage of parameter search algorithms is that a relatively efficient algorithm has been developed for finding a local maximum for the likelihood function. This algorithm, called the inside-outside algorithm (Baker 1979; Lari and Young 1990), begins with some initial setting of the rule probabilities and re-estimates these probabilities on a corpus until a local maximum of the corpus likelihood is reached. While this seems encouraging at first, attempts to induce grammars using this algorithm (Carroll and Charniak 1992) proved disappointing. One reason for failure which the authors propose is that the algorithm tends to converge to local maxima which are not good grammars. A different reason, suggested in Klein and Manning (2002), is a poor choice of the set of possible grammar rules in these experiments. Later experiments (Pereira and Schabes 1992; Schabes et al. 1993) showed that this algorithm works successfully when it is trained on bracketed sentences, but no successful application of the algorithm to the induction of PCFGs from unannotated text is known to me.

Non-CFG Syntactic Induction

Despite some slow progress, the performance of algorithms which induce a context free grammar (probabilistic or not) remains disappointing. While some algorithms were reported to successfully induce toy grammars, none seemed to succeed on the task when confronted with real linguistic data. The standard syntactic task in computational linguistics is parsing and it is therefore reasonable to evaluate grammar induction algorithms on the parsing accuracy they achieve. Even when algorithms were able to produce some output on real language input, the accuracy of the parses remained low.

In the last decade, new induction algorithms have been proposed which no longer rely on context free grammars. Instead, various probabilistic models of syntactic structure are used and induction is performed by searching for the parameters which maximize the likelihood of the corpus data. The parse assigned to a sentence is then simply the structure with the highest probability (given the induced parameters). These algorithms use either a constituency (bracketing) representation of syntactic structure (Klein and Manning 2002; Bod 2006a; Bod 2006b; Bod 2007a) or a dependency (link) representation of syntactic structure (Yuret 1998; Paskin 2002; Klein and Manning 2004; Smith and Eisner 2005; Smith and Eisner 2006).

The probability assigned by these models to a syntactic structure is based on the product of the probabilities assigned to the "building blocks" of the structure. In the case of CCM (Klein and Manning 2002), these building blocks are the constituent and non-constituent sequences of parts-of-speech in the structure as well as the contexts of these sequences. The probability distributions induced by the algorithm then specify the probability of a certain sequence of parts-of-speech (or context) as a constituent or a non-constituent (see figure 1.1 for details). In the case of the different variants of U-DOP (Bod 2006a; Bod 2006b; Bod 2007a), the building blocks are subtrees of the syntactic structure and the probabilities are the probability of using each subtree in a derivation (see figure 1.2 for details). Both these models require the syntactic trees to be binary branching. When dependency models are used, the building blocks are the dependency links and the probability distribution describes the probability of two parts-of-speech (or words, in the case of Yuret 1998) being joined by a link. In DMV, Klein and Manning (2004) also added a probability describing the non-attachment of a head beyond its last argument (see figure 1.1 for details). This model has also been used by Smith and Eisner (2005) and Smith and Eisner (2006) with different likelihood maximization techniques.

Many of these recent algorithms perform significantly better than context free grammar induction algorithms. While no CFG induction algorithm has ever been reported to do better on English than the right-branching heuristic (which simply brackets every word together with all words to its right), many recent algorithms (Klein and Manning 2002; Klein and Manning 2004; Smith and Eisner 2005;

CCM:

S - sentence (part-of-speech sequence): $_0\ NN\ _1\ NNS\ _2\ VBD\ _3\ IN\ _4\ NN\ _5$

B - bracketing (boolean matrix): [[$_0\ NN\ _1\ NNS$] $_2$ [$VBD\ _3$ [$IN\ _4\ NN\ _5$]]]

$B_{ij} = true \iff$ bracket from i to j:

	0	1	2	3	4	5	
		t	t			t	0
			t				1
				t		t	2
					t	t	3
						t	4
							5

α_{ij} - parts-of-speech from i to j (e.g. $\alpha_{02} = NN\ NNS$).
x_{ij} - the context of α_{ij} (e.g. $x_{02} = \diamond - VBD$).

CCM defines a probabilistic model $P(S, B) = P_{bin}(B)P(S|B)$ with P_{bin} a uniform distribution over all binary branching bracketings and

$$P(S|B) = \prod_{i<j} P(\alpha_{ij}|B_{ij})P(x_{ij}|B_{ij})$$

DMV:

Projective dependency structure D of sentence S (see section 4.1 for definitions):

Each dependency d is a link from a head h to a dependent a.

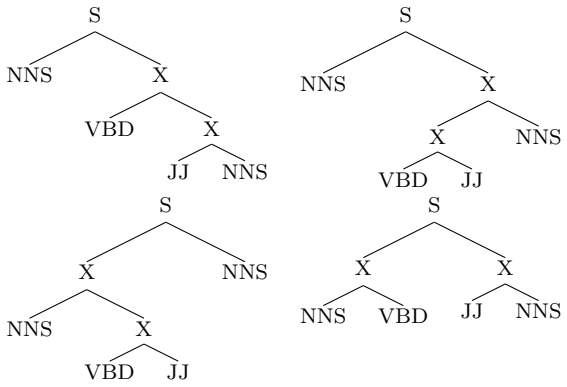
DMV defines the following generative probabilistic model for $P(D, S)$:

$D(h)$ - dependency structure rooted at h ($D = D(root)$).
$deps_D(h, l/r)$ - dependents of h (in D) to the left/right of h.
$adj = true$ iff no dependent has yet been generated in the current direction.

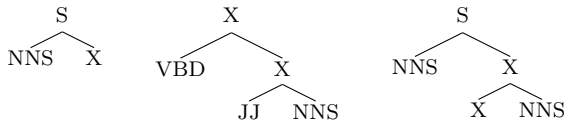
$$P(D(h)) = \prod_{dir \in \{l,r\}} \left(\prod_{a \in deps_D(h,dir)} P_{\text{STOP}}(\neg\text{STOP}|h, dir, adj)\ P(a|h, dir)P(D(a)) \right) \times P_{\text{STOP}}(\text{STOP}|h, dir, adj)$$

Figure 1.1: Klein and Manning's CCM (2002) and DMV (2004) models. The EM algorithm (with the sentences S as observed and bracketing B or dependencies D as unobserved) is used to search for the model parameters which (locally) maximize the likelihood of the (unannotated) corpus. Each sentence is assigned the most probable structure (bracketing/dependency) according to these parameters.

Every sentence (part-of-speech sequence) in the input corpus is assigned all possible binary trees:



All subtrees are extracted:



etc.

Each subtree t in this collection is assigned a probability:

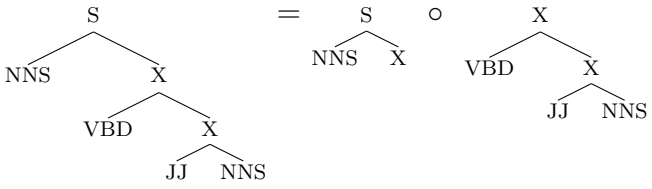
U-DOP: $$P(t) = \frac{|t|}{\sum_{t':r(t')=r(t)} |t'|}$$

where $r(t)$ is the root node of t (S or X) and $|t|$ is the number of times t appears in the subtree collection.

UML-DOP: Expectation maximization beginning with U-DOP's estimates.

U-DOP*: Using the DOP* estimator of Zollmann and Sima'an (2005).

A derivation constructs a tree from subtrees:



The probability of a derivation is the product of the probabilities of the subtrees it uses: $P(t_1 \circ \ldots \circ t_n) = \prod_i P(t_i)$. The probability of a tree is the sum of probabilities of all its possible derivations: $P(T) = \sum_{\{t_1 \circ \ldots \circ t_n = T\}} \prod_i P(t_i)$. In practice, only the most probable derivations are summed.

The parse assigned to a sentence is the tree with the highest probability.

Figure 1.2: Bod's U-DOP (Bod 2006b), UML-DOP (Bod 2006a) and U-DOP* (Bod 2007a) algorithms.

Smith and Eisner 2006; Bod 2006a; Bod 2006b; Bod 2007a) do significantly better than this baseline (see chapter 7 for details).

1.3.5 From Grammar Induction to Parser Induction

The move away from context free grammars has significantly improved the parsing accuracy of unsupervised parsers. While some of this improvement can be attributed to the details of the design or to the use of the expectation maximization technique, I would like to suggest that it is the move from *grammar induction* to *parser induction* which has contributed most to the improvement.

We have seen that the construction of a context free grammar requires two types of decisions to be made: syntagmatic (which sequences of words are constituents) and paradigmatic (which sequences can be substituted for each other). These are two aspects inherent to what we expect from any grammar and neither can be ignored in the process of induction. In contrast, unlabeled parsing, which only requires the parser to identify the constituents (or dependency links) but does not require them to be labeled, is purely syntagmatic (by definition). A parser induction algorithm can therefore focus on learning to detect syntactic units while ignoring substitutability. Indeed, none of the recent successful algorithms (Klein and Manning 2002; Klein and Manning 2004; Bod 2006a; Bod 2006b; Bod 2007a) can determine which constituents are substitutable. Even when contexts are used (as in the CCM algorithm of Klein and Manning 2002) they are only used to determine the probability that the sequence appearing inside the context is a constituent and not to decide which sequences can be substituted for each other. Another example is the memory based algorithm of Dennis (2005), which uses alignment just as in older clustering algorithms but stops short of creating substitution classes. Instead, it directly uses the alignments to make parsing decisions.

In contrast to these parser induction algorithms, grammar induction algorithms need to perform both syntagmatic and paradigmatic induction. In practice, the emphasis was always on the paradigmatic aspect of the induction. This is implied in Stolcke's (1994) comment that syntagmatic merges must usually be followed by paradigmatic merges to produce any improvement on his objective function. This shows that while formally both syntagmatic and paradigmatic relations are learned, it is only the paradigmatic relation which is the driving force behind the induction process. It is not surprising therefore that such algorithms produce poor parsers. The few grammar induction algorithms that did incorporate some explicit mechanism to distinguish constituents from non-constituents (Mori and Nagao 1995; Clark 2001) seem to have gained in parsing accuracy from this. Still, it was only when the focus shifted completely from substitutability to the detection of constituents that parsing accuracy began to improve significantly. Substitutability, the essential idea of the Harris method, which has been seen as a starting point for the induction process for so long, turns out to be unnecessary

in unsupervised parsing.

This does not mean that substitutability is not an important linguistic notion or that grammars are not an important linguistic tool (in generating new sentences, for example) but it does mean that the first step in the learning of syntax, the discovery of the structure of the utterances, can be done without them. Substitutability can then be learned based on this syntactic structure rather than being used to determine it.

This having been said, the notion of substitutability still plays one important role in recent unsupervised parsing algorithms: they all use part-of-speech sequences in place of words as their input (with the exception of Yuret 1998). One may wonder whether this is necessary. In this thesis I suggest that the answer is probably no and I present an unsupervised parser which completely does away with substitutability, even at the word level.

1.4 The Road Taken: Learning to Parse Incrementally

The present thesis is about parser induction. It takes the view that the identification of syntactic units and relations in an utterance does not require the notion of substitution or the definition of a grammar. It makes this explicit by defining a non-deterministic parser and learning a *parsing function* which decides among the various parsing options open to the non-deterministic parser.

When designing an unsupervised parser, it is useful to look at the way humans process language even if one is not interested in cognitive modeling and it is useful to look at the common properties of languages even if one is not looking for a universal grammar. Of the many properties of language and language processing discovered by researchers, I have chosen to make primary use of three: the incrementality of human language processing, the skewness of syntactic tree structures and the Zipfian distribution of words. All these are fundamental and universally accepted properties of language. The use of these properties leads to a greedy parser in which both parsing and learning are local. As a result, learning and parsing are fast, but not at the expense of parsing accuracy, which remains high by current unsupervised parsing standards.

1.4.1 Incrementality

Humans interpret language as it is being heard or read, and do not have to wait for the end of an utterance to determine the structure and meaning of its beginning. This is referred to as the incrementality of human language processing, and has been thoroughly studied by psycholinguists (see e.g. Crocker et al. 2000). While incrementality is widely acknowledged to be a property of human language processing, most grammars are not specifically designed to be applied

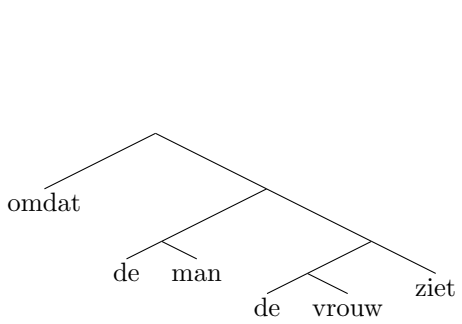
incrementally and most standard parsers are not incremental. Even in a grammatical framework such as combinatory categorial grammar (Steedman 2000), which is supposed to easily accommodate incremental parsing, wide coverage parsers (Hockenmaier and Steedman 2002; Clark and Curran 2004) are not incremental. Thus, in computational linguistics, incrementality is usually seen as an additional burden on the design of a system rather than as a useful tool in its development. But incrementality can be most useful, because it considerably constrains the possibilities a language interpreter has to consider (Church 1980). In the specific case discussed in the present work, this interpreter is a parser which has to determine the syntactic structure of an utterance. While in most standard parsing algorithms the end of the utterance can potentially affect the parse of the beginning of the utterance, this cannot happen in an incremental parser. As a result, an incremental parser has fewer possibilities to consider at each step. This not only restricts the search space for the parser but also simplifies the task of the learning algorithm because the learning algorithm only has to learn to distinguish between the possibilities the parser may choose from.

The problem encountered by incremental parsers is that in some utterances the structure of the beginning of the utterance remains ambiguous until a disambiguating word is reached. This seems to be a problem for incremental parsing, but is actually dependent on the syntactic representation chosen: a structure which is ambiguous in one representation is not necessarily ambiguous in another representation, which may leave the ambiguous feature underspecified until the disambiguating word is reached. Not every ambiguity may be solved in this way and linguists have long been aware of the fact that humans can easily handle some ambiguities while having problems processing others (Bever 1970). Psycholinguists have developed various explanations for this difference between ambiguities and some of these proposals are representational in nature: only the difficult ambiguities are ambiguous in the proposed representations (Weinberg 1993; Weinberg 1995; Gorrell 1995a; Gorrell 1995b; Sturt and Crocker 1996). This will be discussed in section 4.3. In the present work I adopt a similar approach and develop a new link based representation of syntactic structure which is well suited for incremental parsing.

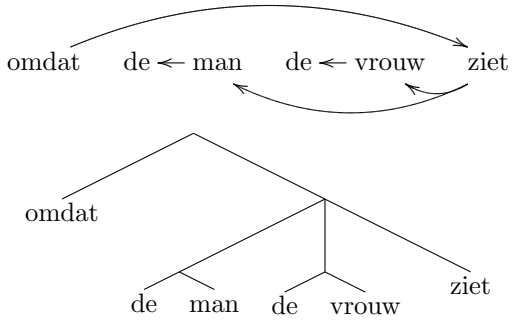
1.4.2 Skewness

The syntactic structure of natural language is skewed. This simply means that when the syntactic structure of an utterance is represented by a tree, each node in the tree has at least one short branch (figure 1.3a). The shorter the shortest branch is, the greater the skewness. In chapter 4, I examine several syntactically annotated corpora to show that a significant degree of skewness can be found in those annotations. The syntactic representation I introduce in this thesis easily captures this skewness.

In contrast, phrase based representations of syntactic structure, such as con-



(a) This syntactic tree of a Dutch phrase is skewed because under every node there is a branch of length at most 2. The shortest branch is sometimes on the left and sometimes on the right.



(b) When the syntactic tree (bottom) is derived from the dependency structure (top) by creating a node for every head and all its direct and indirect dependents, the tree is even more skewed because under every node there is a branch of length 1.

Figure 1.3: An example of skewed syntactic trees.

text free grammars, allow (a-priori) any tree structure and, therefore, a learning algorithm for such representations must discover by itself the skewness property of syntactic trees. However, if this property is indeed universal, there is no need to burden the learning algorithm with its discovery and it is possible to code skewness directly into the parser.

The other extreme is taken by dependency structures (see section 4.1), in which a head word is connected by links to all its dependents (which may, in turn, be heads of other dependents). The straightforward way to construct constituents from a dependency structure is to create for each head word a constituent covering it together with all its direct and indirect dependents (figure 1.3b). The resulting tree is skewed because every head word is attached immediately under the node it heads. This skewness is too strong, however, especially for sentential constructions that combine a subject with a predicate (see the example in figure 1.3 and chapter 4 for details). Therefore, the skewness defined by dependency structures must be relaxed.

The syntactic representation I introduce here is based on links between words, and can easily capture the skewness of syntactic structure in a way similar to that of dependency structures. However, by labeling each link by a number (its *depth*) the representation allows the degree of skewness to be lower than that of dependency structures. I will argue in chapter 4 that the resulting skewness is close to that which is actually observed in natural language.

1.4.3 The Incremental Parser

Having defined a representation for syntactic structure, the next step is to define an efficient parser for that representation, that is, an algorithm which takes an utterance as input and outputs the syntactic structure of that utterance. While the syntactic representation I use was chosen to facilitate incremental parsing, it is the parser I describe which actually implements this incrementality. By doing so, it also defines an exact notion of incrementality (since there are multiple ways of doing so). From now on, I will refer to this simply as the *incremental parser*.

The syntactic formalism used by the parser ensures that the parser can only output skewed syntactic structures, thus eliminating many spurious candidate structures from the search space. The incrementality of the parser further restricts the search space, thus simplifying parsing even further. If the incrementality and skewness coded in the syntactic representation and parser roughly resemble those of natural languages then this reduction of the search space should not come at the expense of the accuracy of the parser.

The basic incremental parsing algorithm is non-deterministic: at each step it specifies a set of links which may be added to the parse, but does not determine which of these links to add. This is not surprising, since different languages require different parsing decisions to be made. Classically, such idiosyncratic properties of a language are coded for the parser by a grammar of the language. In the case of the incremental parser, this is replaced by a *parsing function* which selects, at each step, one of the options available to the parser. It is the parsing function which has to be learned by the induction process. The learning process is simplified if the parsing function only needs to code the idiosyncratic properties of a language and not the universal properties of language parsing. In the present work, skewness and incrementality were coded as universal properties.

1.4.4 Learning and the Zipfian Distribution

To learn the parsing function, the algorithm I present here makes use of the Zipfian distribution of words. Zipf's law states that words in a language obey a power law probability distribution, which roughly means that there is a small number of words which are very frequent and many words which are extremely infrequent. This has often been seen as a curse in computational linguistics, because it means that many words are too infrequent to collect meaningful statistics for. I suggest, however, that one should not see the glass as half empty, but as half full: a relatively small number of frequent words appears almost everywhere and most words are never too far from such a frequent word. The frequent words can therefore guide the parsing and learning process. This is also the principle behind successful part-of-speech induction.

The Zipfian distribution is a property of words, not of parts-of-speech (which cluster many infrequent words, such as nouns, under a single tag). Therefore, in

contrast to most modern syntactic induction algorithms, it is not only possible but also desirable to use the algorithm I present here directly on words and not on part-of-speech sequences. No clustering is performed at any level and the algorithm works entirely locally. Instead of using parts-of-speech, the algorithm labels each side of each word by its neighbors in the text and, recursively, by the labels of these neighbors. Parsing is then directly guided by these labels. Due to the Zipfian distribution of words, high frequency words dominate the lists of labels and parsing decisions for words of similar distribution are guided by the same labels. This not only simplifies the induction process, but also allows much greater flexibility, since the exact label used at each parse step may depend on the parsing context. In addition, the labels on the left and right side of each word may remain independent.

1.4.5 Bootstrapping

The final ingredient in the learning process is bootstrapping. The learning process is nothing more than a simple process of collecting statistics which result from the parsing process: as an utterance is parsed, the parse determines for the learning process which statistics to collect (a somewhat similar idea can be found in Yuret 1998). The statistics of each word are simply collected from the properties of words which are adjacent to it according to the parse. The notion of adjacency depends on the parse assigned to the utterance and will play a central role in the algorithm.

Because learning is merely the collection of statistics resulting from parsing, the learning process is open-ended and additional training text can always be added without having to re-run the learner on previous training data. Learning does not slow parsing much and experiments show that parsing (which is at the rate of thousands of words per second) is slowed down by about 20% when learning is turned on. This means that, potentially, learning can always remain turned on. This is appealing both for engineering purposes and for cognitive modeling.

One risk of using a bootstrapping process, where learning is influenced by what has been learned before, is that incorrect conclusions reached at the beginning of the learning process reinforce themselves through bootstrapping and cannot be gotten rid of. This is similar in some respects to the problem of search algorithms getting stuck at local minima. I will argue (section 6.2.2) that the learning algorithm I propose does not have this problem.

1.5 Organization of this Thesis

The parsing and learning algorithms are described in chapters 2, 3 and 6. Chapter 2 introduces the basic definitions of *common cover links*, the syntactic representation being used, and some of their main properties. The main algorithm in

this chapter (Algorithm 2.6.5) converts common cover link structures into equivalent bracketings. This can be done incrementally, in parallel with parsing and allows the output of the parser to be compared with standard annotation.

Chapter 3 introduces the non-deterministic incremental parsing algorithm (Algorithm 3.3.1) and proves that it can indeed construct every bracketing incrementally. Next, parsing functions are introduced (Definition 3.4.1) and these functions are used to define a deterministic parsing algorithm (Algorithm 3.4.2).

Chapter 6 completes the description of the algorithm. It describes a framework for inducing a parsing function based on a family of greedy parsing functions (Definition 6.1.3). The learning process selects one of the functions in this family based on a statistics update algorithm (Algorithm 6.2.2). This framework leaves some aspects of the algorithm unspecified and section 6.3 specifies a simple instantiation of this framework, given by a lexical update algorithm for learning (Algorithm 6.3.1) and a weight function (section 6.3.2) for the parsing functions.

Chapters 4 and 5 describe the syntactic representation and parser in more detail. Chapter 4 discusses the linguistic properties of the common cover link representation and of the incremental parser. It also discusses in detail the skewness of syntactic structure. Chapter 5 details all the mathematical properties of the representation and the incremental parser and proves all claims made in previous chapters. The chapter is technical and can be skipped in first reading. It was written to be self-contained, so statements (such as definitions, claims and algorithms) given in previous chapters are repeated in this chapter. To make it easier to locate these statements, they are assigned a number in each chapter in which they appear and both numbers are indicated when the statement is made.

Finally, chapter 7 reports on experiments conducted with the algorithm on several real language corpora.

A short description of some of the main contributions of this work was previously published in Seginer (2007).

Bibliography

Abney, S. P. (1989). A computational model of human parsing. *Journal of Psycholinguistic Research 18*(1), 129–144. [73]

Adriaans, P. W., M. Trautwein, and M. R. Vervoort (2000). Towards high speed grammar induction on large text corpora. In G. Hlavac, V. Feffrey, and J. Wiederman (Eds.), *SOFSEM 2000: Theory and practice of Informatics*, Volume 1963 of *Lecture Notes in Computer Science*, pp. 173–186. Springer. [12, 14]

Altenberg, B. (1987). *Prosodic patterns in spoken English: studies in the correlation between prosody and grammar*. Lund University Press. [171]

Anderson, J. R. (1977). Induction of augmented transition networks. *Cognitive Science 1*(2), 125–157. [10]

Baker, J. K. (1979). Trainable grammars for speech recognition. In J. J. Wolf and D. H. Klatt (Eds.), *Speech Communication Papers for the 97th Meeting of the Acoustical Society of America*, pp. 547–550. [16]

Beavers, J. (2003). More heads and less categories: A new look at noun phrase structure. In S. Müllers (Ed.), *Proceedings of the HPSG-2003 Conference, Michigan State University*, pp. 47–67. CSLI Publications. `http://cslipublications.stanford.edu/HPSG/4/`. [67]

Bever, T. G. (1970). The cognitive basis for linguistic structures. In J. R. Hayes (Ed.), *Cognition and the Development of Language*, pp. 279–362. Wiley. [22]

Bod, R. (2006a). An all-subtrees approach to unsupervised parsing. In *Proceedings of the International Conference on Computational Linguistics/Association for Computational Linguistics (COLING-ACL 2006)*, pp. 865–872. [8, 17, 19, 20, 151, 178, 179, 181, 183, 184, 190]

Bod, R. (2006b). Unsupervised parsing with U-DOP. In *Proceedings of the 10th Conference on Natural Language Learning*, pp. 85–92. [8, 17, 19, 20, 151, 178, 179, 181, 183, 190]

Bod, R. (2007a). Is the end of supervised parsing in sight? In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 400–407. [17, 19, 20, 190]

Bod, R. (2007b). Unsupervised syntax-based machine translation: The contribution of discontiguous phrases. In *Proceedings of Machine Translation Summit XI*. [190]

Carroll, G. and E. Charniak (1992). Two experiments on learning probabilistic dependency grammars from corpora. In C. Weir, S. Abney, R. Grishman, and R. Weischedel (Eds.), *Working Notes of the Workshop Statistically-Based NLP Techniques*, Menlo Park, California, pp. 1–13. AAAI Press. [16]

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press. [4]

Chomsky, N. (1995). *The Minimalist Program*. MIT Press. [204]

Chomsky, N. and H. Lasnik (1993). The theory of principles and parameters. In J. Jacobs, A. von Stechow, W. Sternefeld, and T. Vannemann (Eds.), *Syntax: An International Handbook of Contemporary Research*, pp. 506–569. Walter de Gruyter. Reprinted in Chomsky (1995). [4]

Church, K. (1980). On parsing strategies and closure. In *Proceedings of the 18nd Annual Meeting of the Association for Computational Linguistics*, pp. 107–111. [22]

Clark, A. (2000). Inducing syntactic categories by context distribution clustering. In *Proceedings of the 4th Conference on Natural Language Learning*, pp. 91–94. [12, 150]

Clark, A. (2001). Unsupervised induction of stochastic context-free grammars using distributional clustering. In *Proceedings of the 5th Conference on Natural Language Learning*, pp. 105–112. [12, 13, 14, 20]

Clark, S. and J. R. Curran (2004). Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 103–110. [22]

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. Ph. D. thesis, University of Pennsylvania. [154]

Cook, C. M., A. Rosenfeld, and A. R. Aronson (1976). Grammatical inference by hill climbing. *Informational Sciences (now Information Sciences) 10*, 59–80. [12, 14, 15]

Crocker, M. W., M. Pickering, and C. Clifton (2000). *Architectures and Mechanisms for Language Processing*. Cambridge University Press. [21, 47]

Croft, W. (1995). Intonation units and grammatical structure. *Linguistics 33*, 839–882. [171]

de la Higuera, C. (2005). A bibliographical study of grammatical inference. *Pattern Recognition 38*(9), 1332–1348. [6]

Deacon, T. W. (1997). *The Symbolic Species: The Co-evolution of Language and the Brain*. W.W. Norton. [3]

Dennis, S. (2005). An exemplar-based approach to unsupervised parsing. In *Proceedings of CogSci 2005*. [8, 20]

Dudau-Sofronie, D., I. Tellier, and M. Tommasi (2001). From logic to grammar via types. In L. Popelínský and M. Nepil (Eds.), *Proceedings of the Third Learning Language in Logic Workshop*, pp. 35–46. [10]

Gleitman, L. R. (1990). The structural sources of verb meanings. *Language Acquisition 1*(1), 3–55. [9]

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

Goodall, G. (1987). *Parallel Structures in Syntax: Coordination, Causatives and Restructuring*. Cambridge University Press. [77, 83]

Gorrell, P. (1995a). Japanese trees and the garden path. In R. Mazuka and N. Nagai (Eds.), *Japanese Sentence Processing*, pp. 331–350. Lawrence Erlbaum Associates. [22, 74]

Gorrell, P. (1995b). *Syntax and Parsing*. Cambridge University Press. [22, 74]

Gregory, M., M. Johnson, and E. Charniak (2004). Sentence-internal prosody does not help parsing the way punctuation does. In D. M. Susan Dumais and S. Roukos (Eds.), *HLT-NAACL 2004: Main Proceedings*, Boston, Massachusetts, USA, pp. 81–88. Association for Computational Linguistics. [170, 171]

Haegeman, L. M. V. (1994). *Introduction to Government and Binding Theory* (2nd ed.). Blackwell. [29, 65, 67, 196]

Hamburger, H. and K. Wexler (1975). A mathematical theory of learning transformational grammar. *Journal of Mathematical Psychology 12*(2), 137–177. [10]

Harris, Z. (1946). From morpheme to utterance. *Language 22*(3), 161–183. [8, 11]

Hays, D. G. (1964). Dependency theory: A formalism and some observations. *Language 40*(4), 511–525. [62]

Hockenmaier, J. and M. Steedman (2002). Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 335–342. [22]

Hoekstra, H., M. Moortgat, B. Renmans, M. Schouppe, I. Schuurman, and T. van der Wouden (2003). CGN Syntactische Annotatie. `http://lands.let.kun.nl/cgn/doc_Dutch/topics/version_1.0/annot/syntax/syn_prot.pdf`. [61]

Huck, G. and J. Goldsmith (1996). *Ideology and Linguistic Theory*. London: Routledge. [176]

Hudson, R. A. (1987). Zwicky on heads. *Journal of Linguistics 23*, 109–132. [68, 69]

Hudson, R. A. (1990). *English Word Grammar*. Basil Blackwell. [60, 62, 77]

Hudson, R. A. (2003). An encyclopedia of english grammar and word grammar. online encyclopedia. `http://www.phon.ucl.ac.uk/home/dick/enc-gen.htm`. [62, 83]

Ingram, D. (1989). *First Language Acquisition: Method, Description and Explanation*. Cambridge University Press. [3]

Johnson, K. (2004). Gold's theorem and cognitive science. *Philosophy of Science 71*, 571–592. [6]

Jurafsky, D. and J. H. Martin (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall. [11]

Kanazawa, M. (1998). *Learnable Classes of Categorial Grammars*. CSLI Publication. [6]

Kirby, S. and J. R. Hurford (2002). The emergence of linguistic structure: an overview of the iterated learning model. In A. Cangelosi and D. Parisi (Eds.), *Simulating the Evolution of Language*, pp. 121–148. Springer. [3]

Klein, D. (2005). *The Unsupervised Learning of Natural Language Structure*. Ph. D. thesis, Stanford University. [170, 178, 180, 185, 187, 190, 191]

Klein, D. and C. D. Manning (2002). A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 128–135. [8, 16, 17, 18, 20, 178, 179, 182, 190]

Klein, D. and C. D. Manning (2004). Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pp. 478–485. [8, 17, 18, 20, 63, 151, 154, 178, 179, 181, 183, 185, 187, 190]

Lamb, S. M. (1961). On the mechanization of syntactic analysis. In *1961 Conference on Machine Translation of Languages and Applied Language Analysis (National Physical Laboratory Symposium No. 13)*, Volume II, pp. 674–685. Her Majesty's Stationery Office, London. [6, 12]

Landau, B. and L. R. Gleitman (1985). *Language and Experience: Evidence from the Blind Child*. Harvard University Press. [9]

Langacker, R. W. (1987). *Foundations of Cognitive Grammar Vol. 1: Theoretical Prerequisites*. Stanford University Press. [5]

Langacker, R. W. (1991). *Foundations of Cognitive Grammar Vol. 2: Descriptive Application*. Stanford University Press. [5]

Lari, K. and S. J. Young (1990). The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language 4*, 35–56. [16]

Lecerf, Y. (1960). Programme des conflits, modèle des conflits. *La traduction automatique : bulletin trimestriel de l'Association pour l'Étude et le Développement de la Traduction Automatique et de la Linguistique Appliquée (ATALA) 1*(4,5), 11–20,17–36. [62]

Lee, L. (1996). Learning of context-free languages: a survey of the literature. Technical Report TR-12-96, Harvard University, Center for Research in Computing Technology. [6]

Lewis, R. L. (1993). *An Architecturally-based Theory of Human Sentence Comprehension*. Ph. D. thesis, Carnegie Mellon University. [73]

Marcus, M. P., D. Hindle, and M. M. Fleck (1983). D-theory: Talking about talking about trees. In *Proceedings of the 21st annual meeting on Association for Computational Linguistics*, Morristown, NJ, USA, pp. 129–136. Association for Computational Linguistics. [73]

Marcus, M. P., B. Santorini, and M. A. Marcinkiewicz (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguisitics 19*(2), 313–330. [178]

McDonald, R., K. Crammer, and F. Pereira (2005, June). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, Ann Arbor, Michigan, pp. 91–98. Association for Computational Linguistics. [63]

Mel'čuk, I. A. (1988). *Dependency Syntax: Theory and Practice*. Albany State University of New York Press. [60, 61, 83]

Moltmann, F. (1992). *Coordination and Comparatives*. Ph. D. thesis, Massachusetts Institute of Technology. [77, 83]

Mori, S. and M. Nagao (1995). Parsing without grammar. In *Proceedings of the 4th International Workshop on Parsing Technologies*, pp. 174–185. [12, 13, 14, 20]

Muadz, H. (1991). *Coordinate Structures: A Planar Representation*. Ph. D. thesis, University of Arizona. [77, 83]

Nivre, J. (2004). Incrementality in deterministic dependency parsing. In F. Keller, S. Clark, M. Crocker, and M. Steedman (Eds.), *Proceedings of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*, Barcelona, Spain, pp. 50–57. Association for Computational Linguistics. [49]

Nivre, J. and M. Scholz (2004, Aug 23–Aug 27). Deterministic dependency parsing of english text. In *Proceedings of COLING 2004*, Geneva, Switzerland, pp. 64–70. [63]

Oates, T., T. Armstrong, J. Harris, and M. Nejman (2003). Leveraging lexical semantics to infer context-free grammars. In C. de la Higuera, P. W. Adriaans, M. van Zaanen, and J. Oncina (Eds.), *ECML Workshop on Learning Contex-Free Grammars*, pp. 65–76. [10]

Paskin, M. A. (2002). Grammatical bigrams. In T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems*, Volume 14, Cambridge, MA, pp. 91–97. MIT Press. [17]

Pereira, F. and Y. Schabes (1992). Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pp. 128–135. [16]

Pinker, S. (1979). Formal models of language learning. *Cognition 7*(3), 217–283. [12]

Pinker, S. (1996/1984). *Language Learnability and Language Development*. Harvard University Press. [4, 5, 9]

Potter, S. (1950). *Our Language*. Penguin. [68]

Prescher, D. (2005). Head-driven PCFGs with latent-head statistics. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Vancouver, British Columbia, pp. 115–124. Association for Computational Linguistics. [191]

Pritchett, B. L. (1992). *Grammatical Competence and Parsing Performance*. University of Chicago Press. [73]

Sakakibara, Y. (1997). Recent advances of grammatical inference. *Theoretical Computer Science 185*(1), 15–45. [6]

Schabes, Y., M. Roth, and R. Osborne (1993). Parsing the Wall Street Journal with the inside-outside algorithm. In *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics (EACL 93)*, pp. 341–347. [16]

Schütze, H. (1995). Distributional part-of-speech tagging. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics (EACL 95)*, pp. 141–148. [12, 150, 180]

Seginer, Y. (2007). Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 384–391. [26]

Skut, W., B. Krenn, T. Brants, and H. Uszkoreit (1997). An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing ANLP-97*, Washington, DC, pp. 88–95. [178]

Smith, N. A. and J. Eisner (2005). Guiding unsupervised grammar induction using contrastive estimation. In *Proceedings of the IJCAI Workshop on Grammatical Inference Applications*, pp. 73–82. [17]

Smith, N. A. and J. Eisner (2006). Annealing structural bias in multilingual weighted grammar induction. In *Proceedings of the International Conference on Computational Linguistics/Association for Computational Linguistics (COLING-ACL 2006)*, pp. 569–576. [17, 20]

Solan, Z., D. Horn, E. Ruppin, and S. Edelman (2005). Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences 102*, 11629–11634. [12, 13, 14]

Solomonoff, R. J. (1964). A formal theory of inductive inference. *Information and Control 7*, 1–22,224–254. [14]

Steedman, M. (2000). *The Syntactic Process*. MIT Press. [22]

Stolcke, A. (1994). *Bayesian Learning of Probabilistic Language Models*. Ph. D. thesis, University of California at Berkeley. [15, 20]

Sturt, P. and M. W. Crocker (1996). Monotonic syntactic processing: A cross-linguistic study of attachment and reanalysis. *Language and Cognitive Processes 11*(5), 449–492. [22, 72, 74]

Sturt, P., M. J. Pickering, and M. W. Crocker (1999). Structural change and reanalysis difficulty in language comprehension. *Journal of Memory and Language 40*(1), 136–150. [73]

Tellier, I. (1998). Meaning helps learning syntax. In V. Honavar and G. Slutzki (Eds.), *Grammatical Inference, 4th International Colloquium, ICGI-98*, Volume 1433 of *Lecture Notes in Computer Science*, pp. 25–36. Springer. [10]

Tomasello, M. (2003). *Constructing a Language: a Usage-Based Theory of Language Acquisition*. Harvard University Press. [5]

Tomasello, M. (2006). Acquiring linguistic constructions. In D. Kuhn and R. Siegler (Eds.), *Handbook of Child Psychology*, pp. 255–298. New York: Wiley. [5]

Tomasello, M. and E. Bates (2001). *Language Development: The Essential Readings*. Blackwell. [3]

Valiant, L. G. (1984). A theory of the learnable. In *STOC '84: Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, New York, NY, USA, pp. 436–445. ACM Press. [6]

van Zaanen, M. (2000). ABL: Alignment-based learning. In *Proceedings of the 18th International Conference on Computational Linguistics*, pp. 961–967. [12, 14]

Weinberg, A. (1993). Parameters in the theory of sentence processing: Minimal commitment theory goes east. *Journal of Psycholinguistic Research 22*(3), 339–364. [22, 73]

Weinberg, A. (1995). Licensing constraints and the theory of language processing. In R. Mazuka and N. Nagai (Eds.), *Japanese Sentence Processing*, pp. 235–255. Lawrence Erlbaum Associates. [22, 74]

Wolff, J. G. (1982). Language acquisition, data compression and genralization. *Language & Communication 2*(1), 57–89. [12, 14, 15]

Xue, N., F.-D. Chiou, and M. Palmer (2002). Building a large-scale annotated Chinese corpus. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-2002)*, Taipei, Taiwan. [178]

Xue, N. and F. Xia (2000). The bracketing guidelines for the Penn Chinese treebank (3.0). Technical Report 00-08, IRCS. [78]

Yuret, D. (1998). *Discovery of Linguistic Relations Using Lexical Attraction*. Ph. D. thesis, MIT. [17, 21, 25]

Zollmann, A. and K. Sima'an (2005). A consistent and efficient estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics 10*(2/3), 367–388. [19]

Zuidema, W. (2003). How the poverty of the stymulus solves the poverty of the stymulus. In S. Becker, S. Thrun, and K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15 (Proceedings of NIPS'02)*, pp. 51–58. MIT Press. [3]

Zwicky, A. M. (1985). Heads. *Journal of Linguistics 21*, 1–29. [69]

Introduction to Elementary Probability Theory and Formal Stochastic Language Theory

Rens Bod

1. Introduction

For a book on probabilistic approaches to a scientific discipline, it may seem unnecessary to start with an introduction to probability theory. The reader interested in probabilistic approaches will usually have a working knowledge in probability theory and would directly read the more specialized papers. However, the situation is somewhat different for linguistics. Since probability theory does not form part of a traditional linguistics curriculum, the area of probabilistic linguistics may not be as accessible as some other areas. This is further reinforced by the disciplinary gap between probabilistic and categorical approaches, the first being dominant in psycholinguistics and natural language processing, the second in generative linguistics. One of the goals of this book is to show that these two apparently opposing methodologies go very well together: while categorical approaches focus on the endpoints of distributions of linguistic phenomena, probabilistic approaches focus on the gradient middle ground. That linguistic phenomena *are* gradient, will not be discussed here, as this is extensively shown in the other chapters. But to make these chapters accessible to the linguistics community at large, there is a need to explain the most important concepts from probability theory first. Any additional concept that may be encountered later can be looked up in the glossary. I will only assume that the reader has some elementary knowledge of set theory (see Partee et al. 1990 for a linguistic introduction).

After a brief introduction to the basics of probability theory, I will show how our working knowledge can be put into practice by developing the concept of *probabilistic grammar*. Probabilistic grammars are at the heart of probabilistic linguistics and the reader must be acquainted with them before turning to the specialized chapters. Since many different probabilistic grammars have been proposed in the literature, there is a need for a theory that creates some order between them, just as *Formal Language Theory* creates order between non-probabilistic grammars. While I will only scratch the surface of a *Formal Stochastic*[1] *Language Theory*, I will show that probabilistic grammars evoke their own stochastic hierarchies.

2. What are Probabilities?

Historically, there have been two interpretations of probabilities: an *objectivist* and a *subjectivist* interpretation. According to the objectivist interpretation, probabilities are real aspects of the world that can be measured by *relative frequencies* of outcomes of experiments. The subjectivist view, on the other hand, interprets probabilities as *degrees of belief* or *uncertainty* of an observer rather than having any external significance.

[1] The word "stochastic" is used as a synonym for "probabilistic", but is especially used when it refers to results generated by an underlying probability function.

These two contrasting interpretations are also referred to as *frequentist* vs. *Bayesian* (from Thomas Bayes, 1764).

There is much to say in favor of an objectivist interpretation of probabilities in linguistics: linguistic events occur with a certain frequency and a large number of psycholinguistic experiments shows that frequency plays a key role in both language comprehension and production (Jurafsky, this volume). The chapters in this book are therefore mostly based on a frequentist interpretation of probability.

Whichever of the two interpretations one prefers, probabilities are numbers between 0 and 1, where 0 indicates impossibility and 1 certainty (one can also use percentages between 0% to 100%, but this is less common). While the subjectivist thus relies on an observer's judgment of a probability, the objectivist measures a probability through an *experiment* or *trial* -- the process by which an observation is made. The collection of *outcomes* or *sample points* for an experiment is usually referred to as the *sample space* Ω. The concept of *event* is defined as any subset of Ω. In other words, an event may be any set of outcomes that result from an experiment. Under the assumption that all outcomes for an experiment are equally likely, the probability P of an event A can be defined as the ratio between the size of A and the size of the sample space Ω. Let $|A|$ be the number of elements in a set A, then:

$$P(A) = |A| / |\Omega| \qquad (1)$$

To start with a simple, non-linguistic example, assume a fair die which is thrown once. What is the chance of getting an even number? The sample space of this trial is

$$\Omega = \{1, 2, 3, 4, 5, 6\}$$

The event of interest is the subset containing all even outcomes. Let us refer to this event as A:

$$A = \{2, 4, 6\}$$

Thus the number of elements in A is 3, and the number of elements in Ω is 6; that is, $|A| = 3$ and $|\Omega| = 6$. Then the probability of A is:

$$P(A) = |A| / |\Omega| = 3/6 = 0.5$$

Let us now turn to a slightly more linguistic example: assume a small corpus consisting of 50 unambiguous words of which 25 are nouns, 20 are verbs and 5 are adjectives. Consider the experiment of selecting randomly a word W from this corpus. What is the probability of selecting a verb? The sample space Ω of this trial is the set of all words in the corpus. The event of interest A is the set of verbs, which we may write as $\{W : W \text{ is a verb}\}$. So:

$$P(A) = |A| / |\Omega| = |\{W : W \text{ is a verb}\}| / |\Omega| = 20/50 = 0.4$$

For the sake of brevity, we will often write $P(\{\text{verb}\})$ instead of $P(\{W : W \text{ is a verb}\})$. Thus,

$$P(\{\text{verb}\}) = |\{\text{verb}\}| / |\Omega| = 20/50 = 0.4$$
$$P(\{\text{noun}\}) = |\{\text{noun}\}| / |\Omega| = 25/50 = 0.5$$

$P(\{\text{adjective}\}) = |\{\text{adjective}\}| / |\Omega| = 5/50 = 0.1$

A couple of important observations can now be made. First, note that the probability of selecting either a verb or a noun or an adjective is equal to 1, since in that case the event of interest A is $\{W : W \text{ is any word}\}$, which is equal to the sample space Ω, and thus $P(A) = |\Omega| / |\Omega| = 1$. This corresponds to the intuition that the probability that something will be sampled in this experiment is equal to 1.

Second, note that the *sum* of the probabilities of each event, {verb}, {noun} and {adjective}, is also equal to 1, i.e. 0.4 + 0.5 + 0.1 = 1. If events do not overlap, the probability of sampling either of them is equal to the sum of their probabilities. This is known as the *sum rule*. For example, the probability of selecting either a verb or a noun, usually written as $P(\{\text{verb}\} \cup \{\text{noun}\})$ or $P(\{\text{verb, noun}\})$, is equal to 45/50 = 0.9, which is also equal to the sum $P(\{\text{verb}\}) + P(\{\text{noun}\}) = 0.4 + 0.5 = 0.9$. It is important to note that the event {verb, noun} does *not* refer to the event of a word being in the class of words which can be both a noun *and* a verb. As defined above, events are subsets of the sample space, and {verb, noun} denotes the event of either a noun occurring or a verb occurring.

The two properties we have just noted are actually the rules a so-called *probability function* should obey (plus that it should range over [0,1]). The first rule says that a trial will always produce an event in the event space. That is, the probability that something in the event space will happen, namely $P(\Omega)$, is 1:

$$P(\Omega) = 1 \tag{2}$$

The second rule says that if two or more events do not overlap, the probability that either event occurs is equal to the sum of their probabilities; i.e. for two disjoint events A and B:

$$P(A \cup B) = P(A) + P(B) \tag{3}$$

As long as these rules hold, P is a probability function, also known as a *probability distribution*. (There are some well-studied probability distributions that appear later in this book, such as the binomial distribution and the normal distribution. The glossary gives definitions for these distributions.)

Note that rule (3) can be generalized to any number of events, i.e. for n disjoint events $A_1, A_2, ..., A_n$:

$$P(A_1 \cup A_2 \cup ... \cup A_n) = P(A_1) + P(A_2) + ... + P(A_n) \tag{4}$$

The right-hand side of this sum rule is often conveniently abbreviated by the sum sign Σ:

$$P(A_1 \cup A_2 \cup ... \cup A_n) = \Sigma_i P(A_i) \tag{5}$$

Recall that under the frequentist interpretation, the probability of an event is interpreted as its *relative frequency* in a series of experiments. A classical result from statistics shows that the relative frequency of an event converges to its true probability as the number of experiments increases (*Law of Large Numbers*). Thus, if x is an outcome of some experiment (e.g. throwing a die) and *Count*(x) is the number of times x occurs in N

repeated experiments, then the relative frequency $Count(x) / N$ converges to the probability of x if N goes to infinity. The probability of x is also written as $P(X = x)$, where X is called a *random variable* (see also the glossary).

3. Joint Probabilities and Conditional Probabilities

Let us now extend our notion of simple probability to that of *joint* probability. Joint probabilities are useful if we are interested in events that contain more than one outcome. For example, in an experiment where we randomly sample *two* words from our corpus given in section 1 (rather than just *one* word), what is the probability of an event consisting of a noun and a verb -- given that we sample with replacement[2]? We write this probability as $P(\{noun\} \cap \{verb\})$, or simply as $P(\{noun\},\{verb\})$. We already computed the probabilities of sampling a noun and a verb separately, i.e.:

$P(\{noun\}) = 0.5$
$P(\{verb\}) = 0.4$

Intuitively, this amounts to saying that in 50% of the cases we sample a noun, after which in 40% of the cases we sample a verb. This means that we sample them jointly in 40% of 50%, i.e. in 20% of the cases (in our experiment). Thus, intuitively, the joint probability of sampling a noun and a verb is equal to the product of the probabilities of sampling them separately: $P(\{noun\},\{verb\}) = P(\{noun\}) \times P(\{verb\}) = 0.5 \times 0.4 = 0.2$.[3] We can do this simple multiplication[4] because we designed our experiment in such a way that sampling a verb is independent of having sampled a noun. We say that the events {noun} and {verb} are *independent*. In general, for two independent events A and B:

$P(A, B) = P(A) \times P(B)$ if A and B are independent (6)

It often the case that two events are not independent, i.e. they are *dependent*. We could design an experiment where the probability of sampling a verb changes if we know that we previously sampled a noun. This is for instance the case in an experiment where we sample two *consecutive* words. Suppose that in our corpus 90% of the nouns are followed by verbs. For such an experiment, the probability of sampling a verb given that we first sampled a noun is thus 0.9 (rather than 0.4). This probability is written as $P(\{verb\} \mid \{noun\})$ and is called the *conditional probability* of a verb given that we have seen a noun. But what is now the probability of sampling a noun *and* a verb in this particular experiment? We know that:

$P(\{noun\}) = 0.5$
$P(\{verb\} \mid \{noun\}) = 0.9$

---

[2] In this book and in probabilistic linguistics in general, the word sampling always refers to sampling with replacement.

[3] Note that the probabily of first sampling a verb and then a noun is also 0.2. This is because set intersection is commutative, i.e. $\{noun\} \cap \{verb\} = \{verb\} \cap \{noun\}$ and therefore $P(\{noun\} \cap \{verb\}) = P(\{verb\} \cap \{noun\})$. This also means that the probability of sampling a noun and a verb in *any* order is equal to $0.2 + 0.2 = 0.4$.

[4] In this book, multiplications are often written without the multiplication sign. Thus $P(A) \times P(B)$ is also written as $P(A)P(B)$.

That is, in 50% of the cases we sample a noun, after which in 90% of the cases we sample a verb (in this experiment). This means that we sample them jointly in 90% of 50% of the cases, which is 45% of the cases. Thus, the joint probability $P(\{\text{noun}\},\{\text{verb}\})$ is equal to the product $P(\{\text{noun}\}) \times P(\{\text{verb}\} \mid \{\text{noun}\}) = 0.5 \times 0.9 = 0.45$. In general, for two events A and B:

$$P(A, B) = P(A) \times P(B \mid A) \tag{7}$$

which reads as "the probability of A and B equals the probability of A, times the probability of B given A". Note that this formula generalizes over formula (6): if the events A and B are independent, $P(B \mid A)$ is equal to $P(B)$, and (7) reduces to (6). Formula (7) is generally known as the *multiplication rule* or *product rule*. The product rule can also be written as a general definition for conditional probability:

$$P(B \mid A) = P(A, B) / P(A) \tag{8}$$

Most textbooks on probability theory first define the concept of conditional probability from which next the formula for joint probability is derived. For the current exposition, it seemed more intuitive to me to do this the other way round.

From (8), *Bayes' rule* can be derived. First, write (6) as:

$$P(H \mid E) = P(E, H) / P(E) \tag{9}$$

where, in the context of *Bayesian reasoning*, $P(H \mid E)$ usually reads as the probability of an hypothesis H given some evidence E. Second, since set intersection is commutative (i.e., $A \cap B = B \cap A$), the joint probability $P(E, H)$ is equal to $P(H, E)$, and we can therefore write the right-hand side of (9) also as $P(H, E) / P(E)$, which according to (7) is equal to $P(H) \times P(E \mid H) / P(E)$. Thus, (9) can be written as:

$$P(H \mid E) = P(H) \times P(E \mid H) / P(E) \tag{10}$$

This formula, known as Bayes' rule, is useful if the conditional probability $P(H \mid E)$ is more difficult to compute than $P(H)$ and $P(E \mid H)$. We will see later in this book how Bayes' rule can be applied to linguistic phenomena.

Turning back to the concept of joint probability, the product rule (7) for two events can be generalized to multiple events. For example, the joint probability of three events A, B and C is

$$P(A, B, C) = P(A) \times P(B \mid A) \times P(C \mid A, B) \tag{11}$$

which reads as "the probability of A, B and C equals the probability of A, times the probability of B given A, times the probability of C given A and B". The proof of (11) follows straightforwardly when we combine the Associative property of set intersection (i.e., $A \times B \times C = A \times (B \times C) = (A \times B) \times C$) with formula (7): $P(A, B, C) = P(A, (B, C)) = P(A) \times P(B, C \mid A) = P(A) \times P(B \mid A) \times P(C \mid A, B)$. And for n events $A_1, A_2, ..., A_n$, the multiplication rule becomes:

$$P(A_1, A_2, ..., A_n) = P(A_1) \times P(A_2 \mid A_1) \times ... \times P(A_n \mid A_1, A_2, ..., A_{n-1}) \tag{12}$$

which is also known as the *chain rule*. Remember that in an experiment where the events $A_1, A_2, ..., A_n$ are *independent*, formula (12) simply reduces to:

$$P(A_1, A_2, ..., A_n) = P(A_1) \times P(A_2) \times ... \times P(A_n) \quad (13)$$

Sometimes, each event depends *only* on the directly previously occurring event, in which case formula (12) reduces to:

$$P(A_1, A_2, ..., A_n) = P(A_1) \times P(A_2 \mid A_1) \times ... \times P(A_n \mid A_{n-1}) \quad (14)$$

Formula (14) stands for what is more commonly known as a *first-order Markov model* where each event is dependent only on its preceding event; and formula (13) corresponds to a *zero-order Markov model*. In general, a *k-th order Markov model* assumes that each event is dependent only on a fixed number of k preceding events, where k is called the *history* of the model. For several decades, Markov models were assumed to be inadequate for linguistics because they were applied to word sequences (*n-grams*, such as *bigrams* or *trigrams*) only, without taking into account the grammatical structure of these sequences. Yet, we will see in the following section that formulas (12) through (14) can just as well be applied to grammatical structures.

It is useful to introduce the product sign Π, which abbreviates long products (and is analogous to the sum sign Σ, which abbreviates long sums). For example, (12) is often written as:

$$P(A_1, A_2, ..., A_n) = \prod_i P(A_i \mid A_1, A_2, ..., A_{i-1}) \quad (15)$$

And if the events are independent, (15) reduces to (as with (13)):

$$P(A_1, A_2, ..., A_n) = \prod_i P(A_i) \quad (16)$$

It is important to understand the difference in use between the sum rule in (4) and the product rule in (6) and (7). The sum rule describes the probability that either event A or B occurs in some experiment, which is equal to the *sum* of their probabilities (provided that A and B are disjoint[5]). The product rule, on the other hand, describes the probability that both A and B occur as a joint event in an experiment where events can have more than one outcome; and this probability is equal to the *product* of the probabilities of A and B (or in the general case, to the product of the probability of A and the conditional probability of B given A).

4. Probabilistic Grammars

We have now introduced just enough concepts from probability theory to deal with an example of actual linguistic interest: *probabilistic grammars* (also called *stochastic grammars*). As the reader will see in the following chapters, probabilistic grammars are used to describe the probabilistic nature of a vast number of linguistic phenomena, such as phonological acceptability, morphological alternations, syntactic wellformedness, semantic interpretation, human sentence disambiguation, sociolinguistic variation, etc.

[5] If A and B are not disjoint, there is double counting, which means that the counts of the intersection of A and B should be subtracted. Thus, for the general case: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

One of the most widely-used probabilistic grammars is the *Probabilistic Context-Free Grammar* or *PCFG* (also called *Stochastic Context-Free Grammar*). We will explain a PCFG by a simple example. Suppose we have a very small treebank consisting of only two surface trees for the sentences *Mary hates visiting relatives* and *John likes buzzing bees* (figure 1). We will assume that each tree in the treebank corresponds to the structure as it was perceived for that sentence by some hypothetical natural language user. (We leave out some subcategorizations to keep the example simple.)

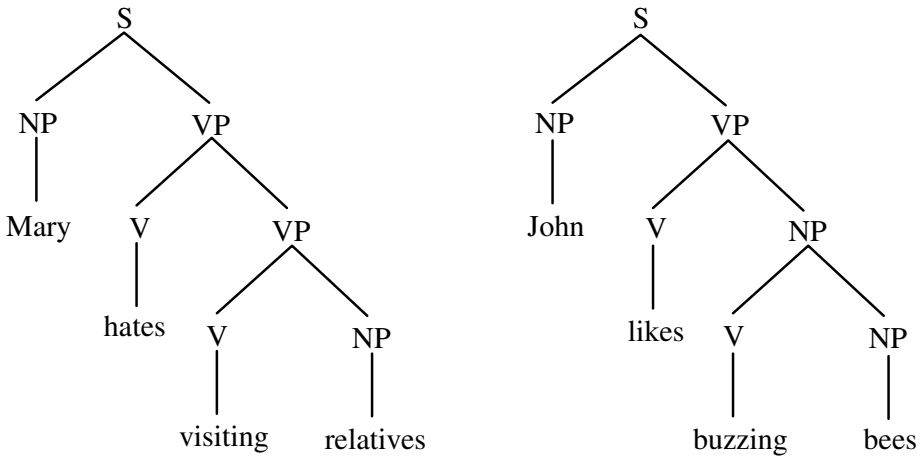


Figure 1. A treebank of two trees

Note that the only difference (apart from the words) between the two structures is the syntactic label covering the last two words of the sentences, which is a VP in the first, and an NP in the second sentence. By reading the rules off the trees, we obtain the context-free grammar (CFG) implicit in these structures. The following table gives these rules together with their frequencies in the treebank.

Rule	**Frequency**
S -> NP VP	2
VP -> V NP	2
VP -> V VP	1
NP -> V NP	1
NP -> Mary	1
NP -> John	1
NP -> relatives	1
NP -> bees	1
V -> hates	1
V -> likes	1
V -> visiting	1
V -> buzzing	1
Total	14

Table 1. The rules implicit in the treebank of figure 1

This table allows us to derive, for example, the probability of the rule S -> NP VP in the treebank. Or more precisely: the probability of randomly selecting S -> NP VP from among all rules in the treebank. The rule S -> NP VP occurs twice in a sample space of 14 rules, hence its probability is 2/14 = 1/7. However, usually we are not so much interested in the probability of a single rule, but rather in the probability of a combination of rules (i.e. a *derivation*) that generates a particular sentence. The grammar derived from the treebank in table 1 generates an infinite number of sentences, including *Mary likes buzzing bees, Mary likes visiting buzzing bees*, *Mary likes visiting buzzing visiting bees* etc. Thus while these sentences are not in the treebank, they can be generated by productively combining fragments from the treebank trees.[6] For example, *Mary likes buzzing bees* can be generated by combining the following rules from table 1:

S -> NP VP, NP -> Mary, VP -> V NP, V -> likes, NP -> V NP, V -> buzzing, NP -> bees.

Figure 2. Treebank-rules for deriving *Mary likes buzzing bees*

This combination of rules, or derivation, produces the following tree structure[7]:

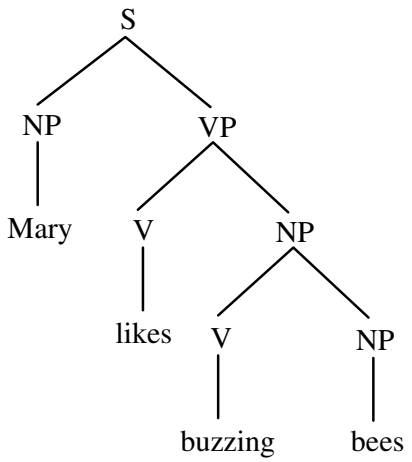


Figure 3. Tree structure produced by the rules in figure 2

Note that the sentence *Mary likes buzzing bees* is ambiguous, i.e., the grammar in table 1 can also produce the following, alternative tree structure for this sentence:

[6] This shows that the "Chomskyan myth" that finite corpora can only generate finite numbers of sentences is fallacious.

[7] Without loss of generality, we will assume that a tree or a sentence is produced by a *leftmost* derivation.

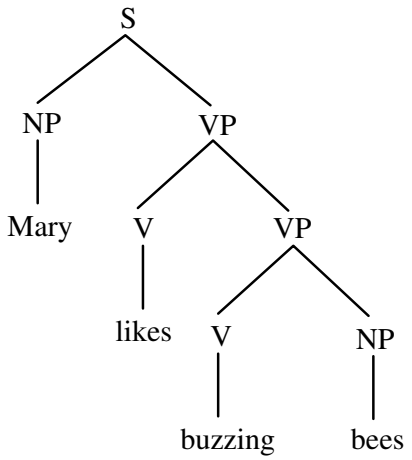


Figure 4. Alternative tree structure generated by the treebank-rules in table 1

One application of probability theory is to provide a ranking of the various tree structures for a sentence by means of their probabilities. How can we determine the probabilities of the two structures in figures 3 and 4? Using the concepts from sections 2 and 3, a tree structure can be seen as an *event* containing the context-free rules in an *experiment* which parses a particular sentence by a (leftmost) derivation. In this experiment, we thus first select an S-rule from among all possible S-rules, then we select the next rule among the rules that can be combined with the previous rule (i.e. which starts with the same category as the first category of the right-hand side of the previous rule), and this is repeated until only terminal leaves remain. Note that this experiment is only well-defined if each rule can indeed be combined with the previous rule and if the first rule starts with an S. Thus the probability of the derivation resulting in the tree of figure 3 is the *joint probability* of:

(1) selecting the rule S -> NP VP among the rules starting with an S,
(2) selecting the rule NP -> Mary among the rules starting with an NP,
(3) selecting the rule VP -> V NP among the rules starting with a VP,
(4) selecting the rule V -> likes among the rules starting with a V,
(5) selecting the rule NP -> V NP among the rules starting with an NP,
(6) selecting the rule V -> buzzing among the rules starting with a V,
(7) selecting the rule NP -> bees among the rules starting with an NP.

Table 2. The probability of a derivation is the joint probability of selecting these rules

The probability of (1) can be computed by dividing the number of occurrences of rule S -> NP VP by the number of occurrences of all rules that start with an S; there are two rules S-> NP VP in the treebank, and the total number of S-rules is also two (in fact they coincide); thus the probability of (1) is 2/2 = 1. Note that this probability is actually the conditional probability P(S -> NP VP | S) and thus the sum of the conditional probabilities of all rules given a certain nonterminal to be rewritten is equal to 1.

The probability of (2) is equal to 1/5 since the rule NP -> Mary occurs once among a total of 5 rules that start with an NP.

The probability of (3) is equal to 2/3 since the rule VP -> V NP occurs twice among a total of 3 rules that start with a VP.

The probabilities of all rules in table 2 are given in table 3:

Event	Probability
(1) selecting the rule S -> NP VP among the rules starting with an S	1
(2) selecting the rule NP -> Mary among the rules starting with an NP	1/5
(3) selecting the rule VP -> V NP among the rules starting with a VP	2/3
(4) selecting the rule V -> likes among the rules starting with a V	1/4
(5) selecting the rule NP -> V NP among the rules starting with an NP	1/5
(6) selecting the rule V -> buzzing among the rules starting with a V	1/4
(7) selecting the rule NP -> bees among the rules starting with an NP	1/5

Table 3. The probabilities of the various rules in table 2

Having computed these probabilities, how can we now compute their joint probability? That is, are the rules to be taken dependent or independent? In other words, should we apply formula (12) or (13)? A crucial assumption underlying Probabilistic Context-Free Grammars is, as for Context-Free Grammars, that the rules in a derivation depend *only* on the nonterminal to be expanded. And this is what we already did in computing the probabilities above by selecting each rule from among the rules that start with the same nonterminal (i.e. we computed the conditional probabilities *P*(S -> NP VP | S), *P*(NP -> Mary | NP) etc. rather than the simple probabilities *P*(S -> NP VP) and *P*(NP -> Mary)). Thus, for a PCFG, the probability of a rule is independent of the derivation it occurs in, and can be computed off-line. Table 4 gives the PCFG-probabilities for all rules that were be derived from the treebank in figure 1 (see table 1).

Rule	PCFG-probability
S -> NP VP	1
VP -> V NP	2/3
VP -> V VP	1/3
NP -> V NP	1/5
NP -> Mary	1/5
NP -> John	1/5
NP -> relatives	1/5
NP -> bees	1/5
V -> hates	1/4
V -> likes	1/4
V -> visiting	1/4
V -> buzzing	1/4

Table 4. PCFG-probabilities for the rules from the treebank in figure 1.

PCFGs can of course be defined independently of how the rule probabilities are "learned". A PCFG which extracts the probabilities directly from a treebank, as shown above, is known as a *Treebank grammar*, which was coined by Charniak (1996) though used before by Bod (1993).

Turning back to the probability of the derivation generating the tree in figure 3, this can now be computed by the product of the probabilities in table 3, that is, $1 \times 1/5 \times 2/3 \times 1/4 \times 1/5 \times 1/4 \times 1/5 = 2/6000 = 1/3000$. This probability is small, reflecting the fact that the grammar produces derivations for infinitely many sentences whose probabilities sum up to 1 only in the limit. But what we are actually interested in is to

compare the probability of this derivation with the probability of the other derivation for *Mary likes buzzing bees* (producing the tree in figure 4). This other derivation consists of the rules:

S -> NP VP, NP -> Mary, VP -> V VP, V -> likes, VP -> V NP, V -> buzzing, NP -> bees.

Figure 5. The rules generating the tree structure in figure 4

and its probability is equal to $1 \times 1/5 \times 1/3 \times 1/4 \times 2/3 \times 1/4 \times 1/5 = 2/3600 = 1/1800$. Thus, the probability of the derivation producing the tree in figure 4 is higher than the probability of the derivation producing the tree in figure 3. Although we must keep in mind that our sample space of two trees is unrealistically small (most available treebanks contain 50,000 trees or more), it is somewhat surprising that figure 4 gets a higher probability than figure 3. We would expect this to be the other way round because the sentence *Mary likes buzzing bees* differs only in one word with the treebank sentence *John likes buzzing bees*, and therefore we may expect a probabilistic grammar to predict that *Mary likes buzzing bees* has as its most probable tree the same tree as associated with the treebank sentence *John likes buzzing bees*, rather than the tree associated with the other treebank sentence *Mary hates visiting relatives*, which differs much more with our input sentence *Mary likes buzzing bees*. However, as said, a crucial assumption underlying PCFGs is that its rules are independent. It is easy to see that this assumption is wrong, even for the subclass of natural language sentences that *are* in fact context-free. For example, the words *buzzing* and *bees* in the NP *buzzing bees* are probabilistically dependent: i.e. the probability of *bees* is not equal to the probability of *bees* given that we have first observed *buzzing*. But this dependency is not captured by a PCFG, since it takes the rules V -> buzzing and NP -> bees as being independent. Thus while a CFG may suffice as a grammar formalism for defining the categorical properties for the context-free subset of sentences, its probabilistic counterpart PCFG does not do the same job for the *non*-categorical properties of this context-free subset.

During the last decade, several alternative models have been proposed that aim to redress the shortcomings of PCFGs. These alternative probabilistic extensions of CFGs have resulted in probabilistic grammars that are provably stronger than PCFGs (we will explain more precisely what we mean by stronger in the following section). One such probablistic grammar makes the probabilities of the rules dependent on the previous rules being used in a derivation, by effectively applying formula (12) to the rules (Black et al. 1993). However, while such a *History-Based Grammar* can thus capture the dependency between *buzzing* and *bees*, it has problems with dependencies between words that are separated by other words, as for example in the sentence *The old man died* where there is a dependency between *old* and *died* but not between *old* and *man*, or *man* and *died*. A History-Based Grammar of the sort in Black et al. (1993) cannot capture this dependency because the rules are made dependent on the *directly* preceding rules, and not on any arbitrary previously used rule(s).

Another probabilistic grammar formalism, which has become quite influential in the field of natural language processing, associates each nonterminal of a context-free rule with its lexical head according to the treebank tree (e.g. Collins 1996; Charniak 1997). However, such a *Head-Lexicalized Probabilistic Grammar* neglects dependencies that go beyond simple headword dependencies, as for example between *nearest* and *to* in the ATIS[8] sentence *Show the nearest airport to Denver*. Since a Head-

[8] Air Travel Information System (see Marcus et al. 1993).

Lexicalized Probabilistic Grammar considers *nearest* to be a non-headword of the NP *the nearest airport*, it incorrectly disambiguates this sentence (it assigns the highest probability to the tree where the PP *to Denver* is attached to *Show*, since the dependency between the headwords *Show* and *to* is more likely in the ATIS treebank than between the headwords *airport* and *to*). A more elaborate discussion about the shortcomings of Head-Lexicalized Probabilistic Grammars is given in Bod (2001b).

What we may learn from these different probabilistic formalisms is that the probability of a *whole* (i.e. a tree) can be computed from the combined probabilities of its *parts*, but that it is difficult to decide what the *relevant* parts are. In a PCFG, the relevant parts are assumed to be the simple CFG rules, which is clearly wrong, while in a head-lexicalized grammar, the parts are assumed to be the rules enriched with their lexical heads, which is also too limited. Another probabilistic grammar formalism, *Probabilistic Tree-Adjoining Grammar* (Schabes 1992; Resnik 1992), takes the elementary trees of a Tree-Adjoining Grammar as the relevant parts (see Bod 1998 for a critique of this formalism).

There is also a formalism that generalizes over most other probabilistic grammars. It does so by taking any subtree (of arbitrary size) as a *part*, including the entire trees from a treebank. This formalism is known as a Data-Oriented Parsing model or DOP model (Bod 1993, 1998), and is formally equivalent to a Probabilistic Tree Grammar. A DOP model captures the previously mentioned problematic dependency between *old* and *died*, or *nearest* and *to*, by a subtree that has the two relevant words as its only lexical items. Moreover, a DOP model can capture arbitrary fixed phrases and idiom chunks, such as *to take advantage of*. Note that a DOP model reduces to a PCFG if the size of the subtrees is limited to the smallest ones.

Let us illustrate with a simple example how a DOP model works. Since the number of subtrees tends usually to be quite large, we will use the following tiny treebank:

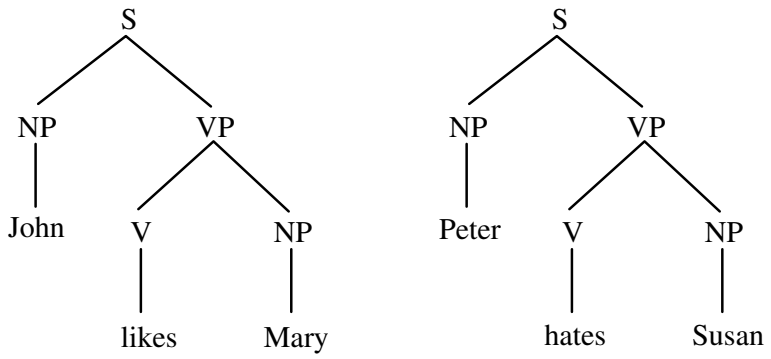


Figure 6. A corpus of two trees

A total of 34 subtrees can be derived from this treebank (at least if we use one specific instantiation of DOP, known as DOP1 or Probabilistic Tree Substitution Grammar -- see Bod (1998)):

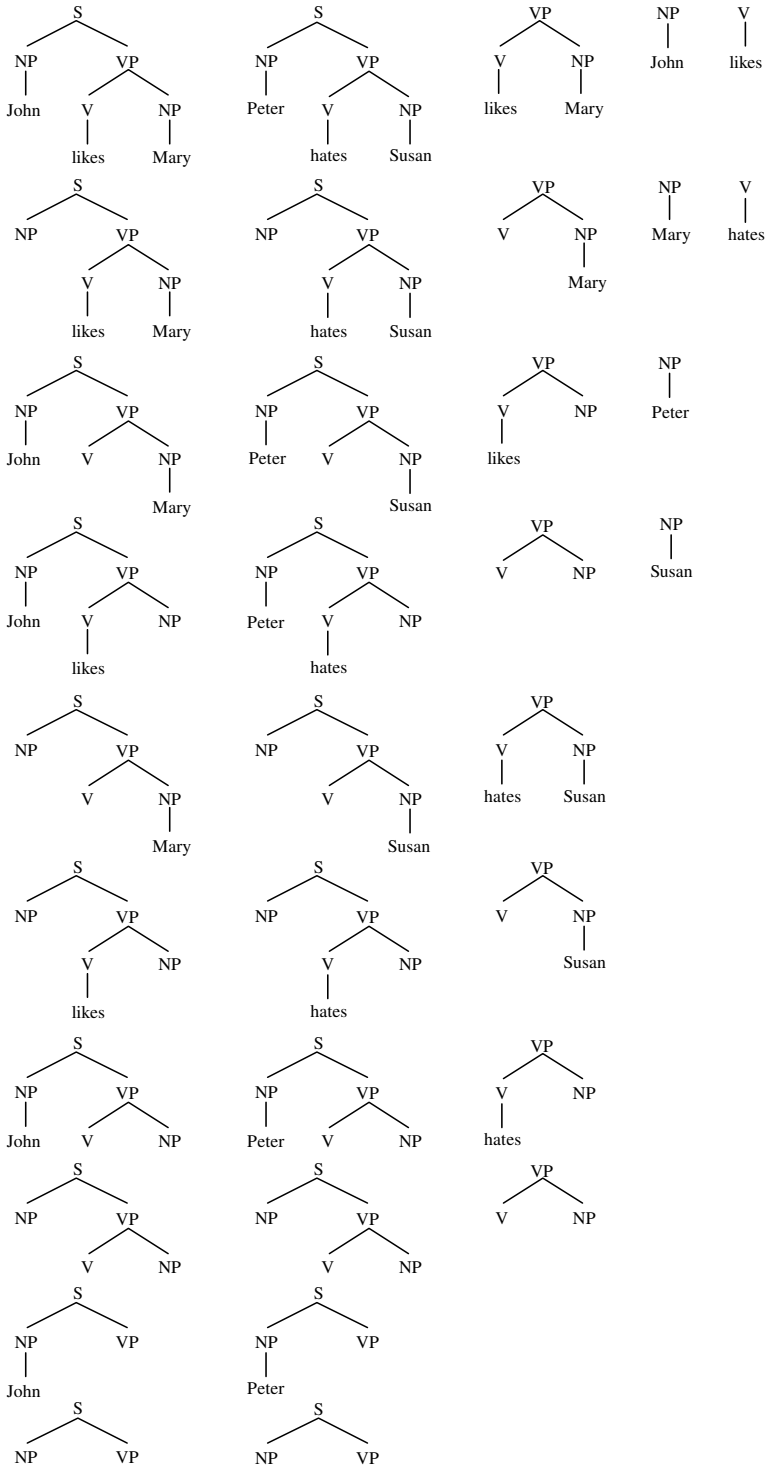


Figure 7. The subtrees derived from the trees in figure 6

Notice that some subtrees occur twice: a subtree may be extracted from different trees, and also from a single tree if the same node configuration appears at different positions.

These subtrees form the underlying grammar by which new sentences are generated. Subtrees are combined using a *node-substitution operation* which is similar to the operation that combines context-free rules in a (P)CFG, and which we indicate by

the symbol "∘". Given two subtrees *T* and *U*, the node-substitution operation substitutes *U* on the leftmost nonterminal leaf node of *T*, written as $T \circ U$. For example, the sentence *Mary likes Susan* can be generated by combining the following three subtrees from figure 7:

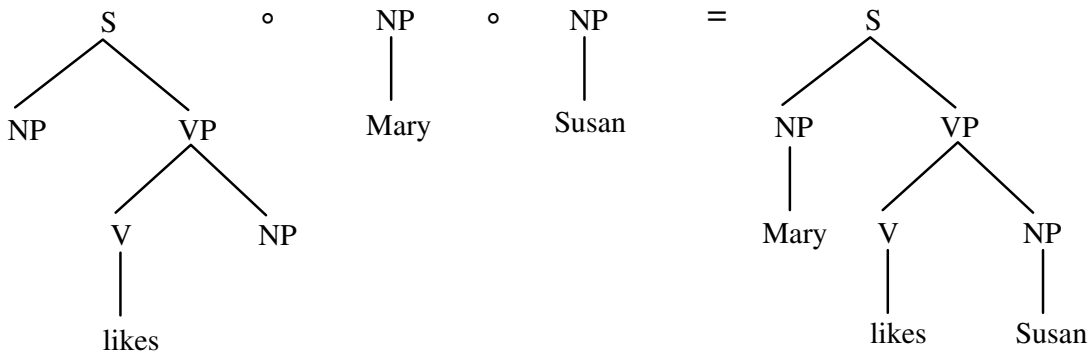


Figure 8. Generating *Mary likes Susan* by combining subtrees

The probability of this derivation is the joint probability of:

(1) selecting the subtree $_{S}$[NP $_{VP}$[$_{V}$[likes] NP]] among the subtrees with root label S,
(2) selecting the subtree $_{NP}$[Mary] among the subtrees with root label NP,
(3) selecting the subtree $_{NP}$[Susan] among the subtrees with root label NP.

Table 5. The probability of a derivation is the joint probability of its subtrees

Thus, the probability of (1) is computed by dividing the number of occurrences of the subtree $_{S}$[NP $_{VP}$[$_{V}$[likes] NP]] in figure 7 by the total number of occurrences of subtrees with root label S: 1/20. The probability of (2) is equal to 1/4, and the probability of (3) is also equal to 1/4.

The probability of the whole derivation is the joint probability of the three selections in table 5. Since each subtree selection is dependent only on the root label and not on the previous selections, the probability of a derivation is, as in PCFG, the product of the probabilities of the subtrees, which is $1/20 \times 1/4 \times 1/4 = 1/320$. Although it is again assumed that the *parts* of our probabilistic grammar are independent, this assumption is now not harmful as in a PCFG, since if any larger subtree occurs in the treebank which includes two (or more) smaller subtrees, it can directly be used as a unit in a derivation thereby taking into account the co-occurrence of the smaller subtrees. This brings us to another feature of DOP: the fact that different derivations can produce the *same* tree. This so-called *spurious ambiguity* may be irrelevant for non-probabilistic grammars, but for probabilistic grammars it leads to a different probability model. For example, the same tree in figure 8 for the sentence *Mary likes Susan* can also be derived by combining the following subtrees:

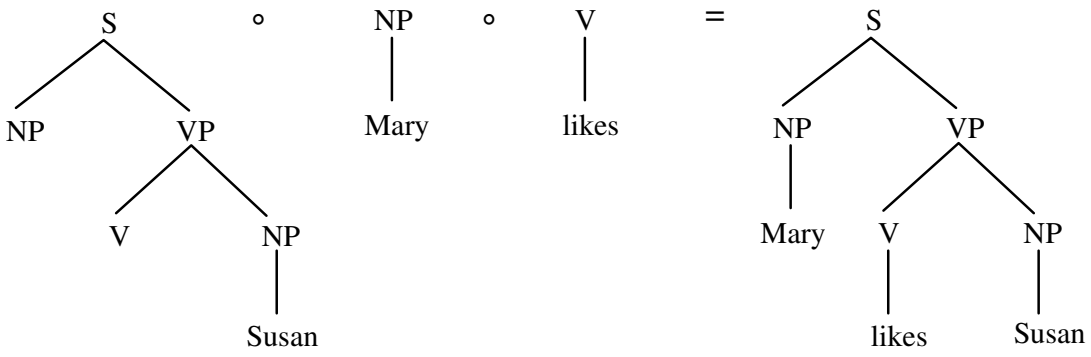


Figure 9. A different derivation, yielding the same parse for *Mary likes Susan*

The probability of this derivation is equal to $1/20 \times 1/4 \times 1/2 = 1/160$, which is different from the probability of the derivation in figure 8, even if it produces the same tree. And there are many more derivations that produce this tree, each with their own probability. The following example is analogous to a PCFG-derivation for *Mary likes Susan*, in that each subtree exactly corresponds to a context-free rewrite rule:

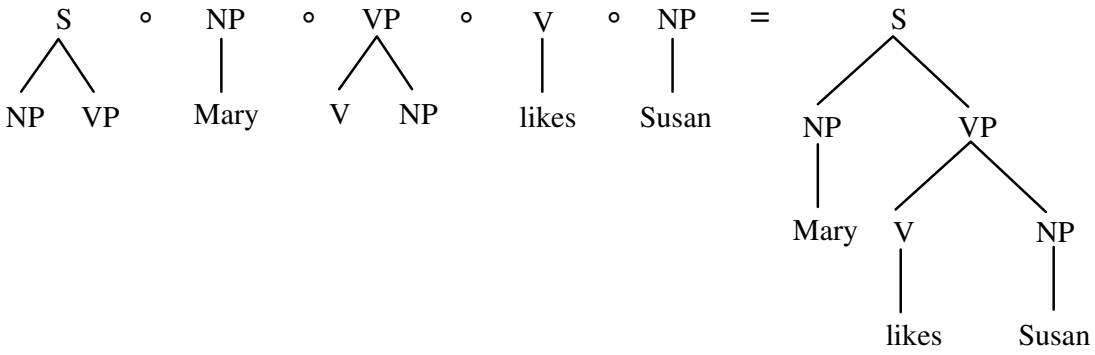


Figure 10. Another derivation, yielding the same parse for *Mary likes Susan*

The probability of this derivation is equal to $2/20 \times 1/4 \times 2/8 \times 1/2 \times 1/4 = 1/1280$, which is again different from the probabilities of the other two derivations generating this tree. Thus in DOP there is not a one-to-one correspondence between derivation and tree as in a PCFG. Instead, for the same tree there may be several distinct derivations. The probability that a certain tree occurs is then the probability that *either* of its derivations occurs. According to rule (4) in section 2, this amounts to saying that the probability of a tree is the *sum* of the probabilities of its derivations (I leave the computation of the tree-probability for *Mary likes Susan* to the reader). Intuitively this means that in DOP there is an accumulation of evidence for a tree: the more derivations a tree has, the larger its probability tends to be. This means that if a tree can be constructed (also) from large parts seen in a treebank, it tends to be ranked higher than a tree which can be constructed only from small subtrees. Note that if we are interested in the probability of generating a certain sentence, we must sum up the probabilities of all different *trees* that generate that sentence -- following the same way of reasoning as for the probability of a tree. It can be shown that the sum of the probabilities of all sentences generated by a DOP model is equal to 1 (following Chi and Geman 1998).

The DOP model explained here is just one of the many DOP models that have been proposed in the literature, and is better known as DOP1 (see Bod et al. 2002 for an overview). The distinctive feature of the general DOP approach, when it was proposed in 1992, is that (1) it directly uses sentence fragments as a grammar, and (2) it does not impose constraints on the size of the fragments. While (1) is now relatively uncontroversial in the field of probabilistic natural language processing (see Manning & Schütze 1999), (2) has not been generally adopted. Many approaches still work either with local trees, i.e. single level rules with limited means of information percolation such as head-words (e.g. Collins 1996; Charniak 1997), or with restricted fragments, as in Probabilistic Tree-Adjoining Grammar, that do not include non-lexicalized fragments (e.g. Schabes 1992; Chiang 2000). However, during the last few years, we can observe a shift towards using more and larger treebank fragments. While initial extensions of PCFGs limited the fragments to the locality of head-words (e.g. Collins 1996; Eisner 1996), later models showed the importance of including context from higher nodes in the tree (e.g. Johnson 1998). The importance of including nonhead-words is now widely accepted (e.g. Collins 1999; Charniak 2000; Goodman 1998). And Collins (2000) argues for "keeping track of counts of arbitrary fragments within parse trees", which has been carried out in Collins and Duffy (2001) who use exactly the same set of all sentence fragments as proposed in the original DOP model by Bod (1992).

From a linguistic point of view, the more interesting question is whether *language users* store sentence fragments in memory, and if they do, whether they store arbitrarily large fragments as proposed by the DOP model. Jurafsky (this volume) reports that people not only store lexical items, but also frequent bigrams (i.e. two-word units), frequent phrases and even whole sentences. For the case of sentences, there is some evidence that language users not only store idioms, but also simple high-frequency sentences such as *I love you* and *I don't know* (Jurafsky, this volume; Bod 2001a). Thus, it seems that language users store sentence fragments in memory and that these fragments can range from two-word units to entire sentences. This suggests that language users need not always generate or parse sentences all over again from the rules of the grammar, but that they can productively reuse previously heard sentences and sentence-fragments. Yet, there is no evidence so far that people store *all* fragments they hear. Only high-frequency fragments seem to be memorized. However, if our language faculty has to *learn* which fragments will be stored, it will initially need to store everything (with the possibility of forgetting them of course), otherwise frequencies can never accumulate. This results in a model which continuously and incrementally updates its fragment memory given new input. We will see that such a model turns out to be important for almost all subfields of (probabilistic) linguistics, ranging from phonology to syntax and from psycholinguistics to sociolinguistics.

Another interesting linguistic question is whether DOP models are too *general*. Since DOP models essentially store all sentences, they do perhaps not provide sufficient constraints for defining the set of possible languages. Since this question is aptly dealt with by Manning (this volume), I will not go into it here. Rather than being too general, DOP models of the sort above are actually too *constrained*, since they have the generative power of context-free languages (this follows from the node-substitution operation for combining subtrees). Although context-free power may suffice for phonology (Pierrehumbert, this volume) and morphology (Baayen, this volume), there are syntactic phenomena, such as long-distance dependencies and cross-serial dependencies, which are known to be beyond context-free. Therefore, a model which is inherently context-free is deemed to be linguistically inadequate. In the last few years, various DOP models have been developed with a generative capacity that is richer than context-free. These DOP models are based on linguistic representations that also allow

for syntactic features, functional categories and semantic forms (cf. Bod & Kaplan 1998; Neumann 1998; Hoogweg 2002). Although a detailed description of these models falls beyond the scope of this chapter, it may be noteworthy that fragments of arbitrary size are indispensable for predicting the correct sentence structure also for these richer DOP models (cf. Bod 1998; Way 1999; Bod & Kaplan 2002). Manning (this volume) goes into some other probabilistic extensions of non-context-free grammars.

5. Formal Stochastic Language Theory

We have seen that a DOP model (or actually DOP1 model) generalizes over a PCFG. But we have not yet shown that DOP is also probabilistically "richer" than a PCFG. That is, we have not proved that it is impossible to create a PCFG for every DOP model. This brings us to the question as to how two probabilistic grammars can be compared. First note that in comparing probabilistic grammars, we are not interested in the traditional notion of generative capacity, since e.g. DOP1, PCFG, History-Based Grammar and Head-Lexicalized Grammar are all context-free. Instead, we are interested in the probability distributions that these probabilistic grammars define over sentences and their trees.

Recall that two of the main concepts in traditional Formal Language Theory are *weak equivalence* and *strong equivalence*. That is, two grammars are said to be weakly equivalent if they generate the same strings, and two grammars are said to be strongly equivalent if they generate the same strings with the same trees. The set of strings generated by a grammar G is also called the *string language* of G, while the set of trees generated by G is called the *tree language* of G.

Analogously, the two main concepts in *Formal Stochastic Language Theory* are *weak stochastic equivalence* and *strong stochastic equivalence*. But before defining these two concepts, we need to introduce the notions of *stochastic string language* and *stochastic tree language*.

The **stochastic string language** generated by a probabilistic grammar G is the set of pairs $<x, P(x)>$ where x is a string from the string language generated by G and $P(x)$ the probability of that string.

The **stochastic tree language** generated by a probabilistic grammar G is the set of pairs $<x, P(x)>$ where x is a tree from the tree language generated by G and $P(x)$ the probability of that tree.

Now we can define *weak* and *strong stochastic equivalence*.

Two probabilistic grammars are called **weakly stochastically equivalent**, iff[9] they generate the same stochastic string language.

Two probabilistic grammars are called **strongly stochastically equivalent**, iff they generate the same stochastic tree language.

Note that if two probabilistic grammars are strongly stochastically equivalent they are also weakly stochastically equivalent.

As an illustration of how Formal Stochastic Language Theory can be used to compare different formalisms, we will investigate whether PCFG and DOP are strongly

[9] The word "iff" stands for "if and only if".

stochastically equivalent (for some other comparisons, see Bod 1998 and Carroll and Weir 2000). Since our instantiation of DOP in this chapter is equal to a Probabilistic Tree-Substitution Grammar (PTSG), we will refer to this DOP model as a PTSG (in accordance with Manning & Schütze 1999: 446-448). Our question is:

Is there a PTSG for which there is a strongly equivalent PCFG but no strongly stochastically equivalent PCFG?

The answer is: yes, there is such a PTSG, and this can be easily shown as follows. Consider the very simple PTSG *G* in figure 11 consisting of three subtrees that are all assigned a probability of 1/3.[10]

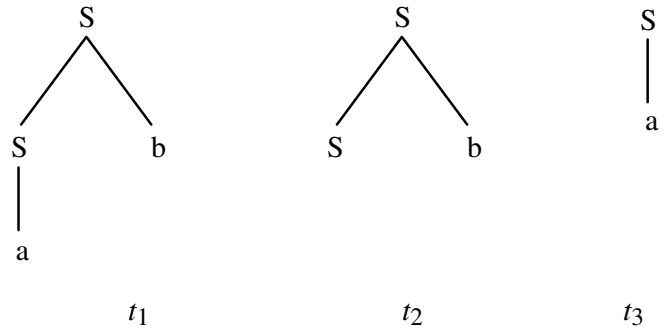


Figure 11. A PTSG consisting of three elementary trees

The string language generated by *G* is {*a*, *ab*, *abb*, *abbb*, *abbbb*, ...} which can be abbreviated as {*ab**}. The only PCFG *G′* which is strongly equivalent with *G* consists of the following productions:

$S \rightarrow Sb$ (1)
$S \rightarrow a$ (2)

G′ would also be strongly stochastically equivalent with *G* if it assigned the same probabilities to the parse trees in the tree language as assigned by *G*. Let us consider the probabilities of two trees generated by *G*, i.e. the trees represented by t_1 and t_3.[11] The tree represented by t_3 has exactly one derivation, which consists of the subtree t_3. The probability of generating this tree is hence equal to 1/3. The tree represented by t_1 has two derivations: by selecting subtree t_1, or by combining the subtrees t_2 and t_3. The probability of generating this tree is equal to the sum of the probabilities of its two derivations, which is equal to $1/3 + (1/3 \times 1/3) = 4/9$.

If *G′* is strongly stochastically equivalent with *G*, it should assign the probabilities 4/9 and 1/3 to (at least) the trees represented by t_1 and t_3 respectively. The tree t_3 is exhaustively generated by production (2); thus the probability of this production should be equal to 1/3: $P(S \rightarrow a) = 1/3$. The tree t_1 is exhaustively generated by applying productions (1) and (2); thus the product of the probabilities of these

[10] This PTSG would correspond to a DOP model of which the subtrees are taken from a treebank consisting only of tree t_1.

[11] Note that the trees t_1 and t_3 are both elements of the set of subtrees of *G* and of the tree language generated by *G*.

productions should be equal to 4/9: $P(S \rightarrow Sb) \times P(S \rightarrow a) = 4/9$. By substitution we get: $P(S \rightarrow Sb) \times 1/3 = 4/9$, from which we derive that $P(S \rightarrow Sb) = 4/3$. This means that the probability of the production $S \rightarrow Sb$ should be larger than 1, which is not allowed. Thus, G' cannot be made strongly stochastically equivalent with G.

What this proof shows is that *there exists a PTSG for which there is no strongly stochastically equivalent PCFG* (even if it's strongly equivalent). On the other hand, one can show that *for every PCFG there exists a strongly stochastically equivalent PTSG*. The proof of the latter is even simpler, because for any rule in any PCFG one can create a minimal one-level subtree (with the same probability) covering exactly the corresponding rule.

Now, if for every PCFG there is a strongly stochastically equivalent PTSG, but not the other way round, then *the set of stochastic tree languages generated by the class of PCFGs is a proper subset of the set of stochastic tree languages generated by the class of PTSGs*. This is what we meant when we said that PTSGs are "richer" than PCFGs.

The goal of this section was to present a framework in which different probabilistic grammars can be compared. The importance of such a comparison should not be underestimated. If we invent a new formalism and next find out that for each grammar in this formalism we can create a strongly stochastically equivalent PCFG, then we haven't made much progress. Thus, rather than being interested in a grammar's place in the Chomsky hierarchy, we are often more interested in its place in the *stochastic* hierarchy within one and the same class of the Chomsky hierarchy.

6. Conclusion

In this chapter I have given the minimum background knowledge to get started with this book. The glossary contains a number of additional concepts that may be encountered in the subsequent chapters. I have only scratched the surface of probability theory and probabilistic grammars. Important topics that have not been touched on include probabilistic regular grammars (which are equivalent to Markov models), probabilistic attribute-value grammars (which generalize over several richer probabilistic grammars) and consistency requirements for probabilistic grammars (which turn our to be particularly interesting for DOP models -- see Bod 2000; Johnson 2002). If the reader feels cheated and wants to know the full picture, then my aim has been achieved. There are excellent textbooks and overview articles on probability theory and formal stochastic language theory, some of which are mentioned below. However, for understanding this book, the current chapter together with the glossary should suffice. At this point, the reader may of course ask whether we really need probabilities and probability theory to cope with gradiency and frequency effects in language, or whether these effects can just as well be accounted for by other approaches such as Optimality Theory (OT) or Connectionism. Then it is really time to dive into the following chapters: probabilistic approaches nowadays cover the entire spectrum of linguistics, and other approaches are increasingly turning to probabilistic models, including OT and Connectionism.

7. Further Reading

There are many good introductory textbooks on probability theory and statistics. A very accessible introduction is Moore and McCabe (1989), which focuses on probability distributions. Other textbooks include Ross (2000) and Feller (1970). For an

introduction from a Bayesian standpoint, see DeGroot (1989). Krenn and Samuelsson (1997) give a tutorial on probability theory aimed at a natural language processing readership. Oakes (1998) gives an overview of the use of statistics in corpus linguistics. More advanced textbooks include Breiman (1973) and Shao (1999). An interesting survey on the emergence of probability in the history of thought is Hacking (1975).

Probabilistic grammars were first studied outside linguistics: Grenander (1967) used probabilistic grammars for pattern recognition, and Booth (1969) studied mathematical properties of probabilistic context-free grammars. Horning (1969) showed that PCFGs can be learned from positive data alone; this result turns out to be quite important for probabilistic linguistics (see Manning, this volume). One of the first papers that argues for PCFGs from a linguistic standpoint is Suppes (1970). Manning and Schütze (1999) give a good overview of the various properties of PCFGs and discuss several enhancements. Jurafsky and Martin (2000) go into the psycholinguistic relevance of PCFGs. Chi and Geman (1998) show that proper probability distributions are obtained if the probabilities of the PCFG-rules are estimated directly from a treebank (as proposed in Bod 1993 and Charniak 1996).

An overview of various probabilistic extensions of CFGs is included in Charniak (1997), Bod (1998, 2001b) and Manning and Schütze (1999). Probabilistic grammars for languages richer than context-free are developed by Abney (1997), Bod and Kaplan (1998) and Johnson et al. (1999), among others. DOP models are covered in Bod (1998) and Bod et al. (2002). For the properties of various probability models for DOP, see Bod (2000), Bonnema (2002), Goodman (2002) and Johnson (2002).

Initial comparisons between different probabilistic grammars focused on their stochastic string languages (e.g. Fu 1974, Levelt 1974, Wetherell 1980). Bod (1993) distinguishes between weak and strong stochastic equivalence, and Bod (1998) uses these concepts to compare different probabilistic extensions of CFGs, suggesting a hierarchy of probabilistic grammars within the classes of the Chomsky hierarchy. Abney et al. (1999) investigate the exact relationship between probabilistic grammars and probabilistic automata. Carroll & Weir (2000) show the existence of a subsumption lattice of probabilistic grammars where PCFG is at the bottom and DOP at the top.

References

Abney, S. 1997. Stochastic Attribute-Value Grammars. *Computational Linguistics* 23(4): 597-618.

Abney, S., D. McAllester and F. Pereira, 1999. Relating Probabilistic Grammars and Automata. *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. 542-549. College Park, Maryland.

Black, E., F. Jelinek, J. Lafferty, D. Magerman, R. Mercer and S. Roukos, 1993. Towards History-Based Grammars: Using Richer Models for Probabilistic Parsing, *Proceedings ACL'93*, Columbus, Ohio.

Bod, R. 1992. Data-Oriented Parsing. *Proceedings COLING*, Nantes, France.

Bod, R. 1993. Using an Annotated Corpus as a Stochastic Grammar. *Proceedings 6th Conference of the European Chapter of the Association for Computational Linguistics (ACL)*. Utrecht, The Netherlands.

Bod, R. 1998. *Beyond Grammar*. CSLI Publications, The University of Chicago Press.

Bod, R. 2000. Combining Semantic and Syntactic Structure for Language Modeling. *Proceedings of the Eighth International Conference on Spoken Language Processing (ICSLP'2000)*, Beijing, China.

Bod, R. 2001a. Sentence Memory: the Storage vs. Computation of Frequent Sentences. *Proceedings CUNY-2001 Conference on Sentence Processing*, Philadelphia, Pennsylvania.

Bod, R. 2001b. What is the Minimal Set of Fragments that Achieves Maximal Parse Accuracy? *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*. Toulouse, France.

Bod, R. and R. Kaplan, 1998. A Probabilistic Corpus-Driven Model for Lexical Functional Analysis, *Proceedings COLING'98 & ACL'98*, Montreal, Canada.

Bod, R. and R. Kaplan, 2002. A Data-Oriented Parsing Model for Lexical-Functional Grammar. In Bod et al. (2002).

Bod, R., R. Scha and K. Sima'an (eds.) 2002, *Data-Oriented Parsing*. CSLI Publications, The University of Chicago Press. (to appear)

Bonnema 2002. Probability Models for Data-Oriented Parsing. In Bod et al. (2002).

Booth, T. 1969. Probabilistic Representation of Formal Languages. *Tenth Annual IEEE Symposium on Switching and Automata Theory*.

Breiman, L. 1973. *Statistics with a view toward applications*. Houghton Mifflin, Boston.

Carroll, J. and D. Weir, 2000. Encoding Frequency Information in Lexicalized Grammars. In H. Bunt and A. Nijholt (eds.), *Advances in Probabilistic and Other Parsing Technologies*, Kluwer Academic Publishers.

Charniak, E. 1996. Treebank Grammars, *Proceedings AAAI'96*, Menlo Park, Ca.

Charniak, E. 1997. Statistical Techniques for Natural Language Parsing. *AI Magazine*, Winter 1997.

Charniak, E. 2000. A Maximum-Entropy-Inspired Parser. *Proceedings ANLP-NAACL'2000*, Seattle, Washington.

Chi, Z. and S. Geman, 1998. Estimation of Probabilistic Context-Free Grammars. *Computational Linguistics* 24: 299-305.

Chiang, D. 2000. Statistical parsing with an automatically extracted tree adjoining grammar. *Proceedings ACL'2000*, Hong Kong, China.

Collins, M. 1996. A new statistical parser based on bigram lexical dependencies. *Proceedings ACL'96*, Santa Cruz, Ca.

Collins, M. 1999. *Head-Driven Statistical Models for Natural Language Parsing*, PhD thesis, University of Pennsylvania, PA.

Collins, M. 2000. Discriminative Reranking for Natural Language Parsing. *Proceedings ICML-2000*, Stanford, Ca.

Collins, M. and N. Duffy, 2001. Convolution Kernels for Natural Language. *Proceedings Neural Information Processing Systems 2001 (NIPS 2001)*, Alberta, Canada.

DeGroot, M. 1989. *Probability and Statistics*. Addison-Wesley.

Eisner, J. 1996. Three new probabilistic models for dependency parsing: an exploration. *Proceedings COLING-96*, Copenhagen, Denmark.

Feller, W. 1970. *An Introduction to Probability Theory and its Applications*. John Wiley & Sons, New York.

Fu, K. 1974. *Syntactic Methods in Pattern Recognition*. Academic Press, London.

Goodman 1998. *Parsing Inside-Out*. PhD Thesis. Harvard University.

Goodman, J. 2002. PCFG Reductions of Data-Oriented Parsing. In Bod et al. (2002).

Grenander, U. 1967. Syntax-Controlled Probabilities. *Technical Report, Division of Applied Mathematics*, Brown University, Providence, R.I.

Hacking, I. 1975. *The Emergence of Probability*. Cambridge University Press, Cambridge.

Hoogweg, L. 2002. Extending DOP with the Insertion Operation. In Bod et al. (2002).

Horning, J. 1969. *A Study of Grammatical Inference*. PhD Thesis, Stanford University.

Johnson, M. 1998. PCFG Models of Linguistic Tree Representations, *Computational Linguistics* 24(4), 613-632.

Johnson, M. 2002. The DOP1 estimation method is biased and inconsistent. *Computational Linguistics* (in press).

Johnson, M., S. Geman, S. Canon, Z. Chi and S. Riezler, 1999. Estimators for Stochastic Unification-Based Grammars, *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. College Park, Maryland.

Jurafsky D. and J. Martin, 2000. *Speech and Language Processing*. Prentice Hall, NJ.

Krenn, B. and C. Samuelsson 1997. The Linguist's Guide to Statistics. Manuscript, University of Saarbrücken.

Levelt, W. 1974. *Formal Grammars in Linguistics and Psycholinguistics (vol.I)*. Mouton, The Hague.

Manning, C. and H. Schütze, 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge.

Marcus, M., B. Santorini and M. Marcinkiewicz, 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics* 19(2).

Moore, D. and G. McCabe, 1989. *Introduction to the Practice of Statistics*. Freeman, New York.

Neumann, G. 1998. Automatic Extraction of Stochastic Lexicalized Tree Grammars from Treebanks, *Proceedings of the 4th Workshop on Tree-Adjoining Grammars and Related Frameworks*, Philadelphia, PA.

Oakes, M., 1998. *Statistics for Corpus Linguistics*. Edinburgh University Press, Edinburgh.

Partee, B., A. ter Meulen and R. Wall, 1990. *Mathematical Methods in Linguistics*. Kluwer Academic Publishers.

Resnik, P. 1992. Probabilistic Tree-Adjoining Grammar as a Framework for Statistical Natural Language Processing. *Proceedings COLING'92*, Nantes, France.

Ross, S. 2000. *Introduction to Probability Models*. Academic Press, Burlington.

Schabes, Y. 1992. Stochastic Lexicalized Tree-Adjoining Grammars. *Proceedings COLING'92*, Nantes, France.

Shao, J. 1999. *Mathematical Statistics*. Springer Verlag, New York.

Suppes, P. 1970. Probabilistic Grammars for Natural Languages. *Synthese* 22: 95-116.

Way, A. 1999. A Hybrid Architecture for Robust MT using LFG-DOP. *Journal of Experimental and Theoretical Artificial Intelligence* 11: 201-233 (Special Issue on Memory-Based Language Processing).

Wetherell, C. 1980. Probabilistic Languages: A Review and Some Open Questions, *Computing Surveys*, 12(4).

Bayesian Model Merging for Unsupervised Constituent Labeling and Grammar Induction

Gideon Borensztajn
Institute for Logic, Language and Computation, University of Amsterdam
Plantage Muidergracht 24
1018 TV, Amsterdam, the Netherlands
gideon@science.uva.nl

Willem Zuidema
Institute for Logic, Language and Computation, University of Amsterdam
Plantage Muidergracht 24
1018 TV, Amsterdam, the Netherlands
jzuidema@science.uva.nl

Abstract

Recent research on unsupervised grammar induction has focused on inducing accurate bracketing of sentences. Here we present an efficient, Bayesian algorithm for the unsupervised induction of syntactic categories from such bracketed text. Our model gives state-of-the-art results on this task, using gold-standard bracketing, outperforming the recent semi-supervised approach of (Haghighi and Klein, 2006), obtaining an F_1 of 76.8% (when appropriately relabeled). Our algorithm assigns comparable likelihood to unseen text as the treebank PCFG. Finally, we discuss the metrics used and linguistic relevance of the results.

1 Introduction

If the unsupervised induction of grammar is still a key open problem in machine learning, it is not for lack of trying: at least since (Solomonoff, 1964), much talent and effort has been invested in finding algorithms for learning syntactic structure from plain text or semantically or phonetically enriched input data. Almost every combination of reasonable choices for syntactic formalism, search procedure, success criterion and input data has been tried, but little of this work is remembered today. Useful reviews from 3 past decades are (Pinker, 1979; Angluin and Smith, 1983; Sakakibara, 1997). In the last 5 years, new efforts have been made to evaluate such algorithms on manually annotated corpora such as ATIS (Hemphill et al., 1990) and the Penn WSJ corpus (Marcus et al., 1993). An important breakthrough was the CCM algorithm of (Klein and Manning, 2002a), which assigns brackets to sentences in a corpus and was the first to outperform a right-branching base-line onm English. Since then, several other algorithms have been described that also score better than this base-line (Klein and Manning, 2004; Dennis, 2005; Bod, 2006; Seginer, 2007). These models have in common that they do not label constituents with syntactic categories and do not use a generative, PCFG probability model[1].

Although accurate bracketing is important, it is clear that it is only a first, intermediate step. For cognitive plausibility, as well as for most NLP applications, we need at least an account of how constituents are categorized. This task is similar to POS-tagging, and because the latter can be done very accurately with existing techniques, the former is often assumed to be an easy task too. Therefore, it has not received as much attention as it deserves. However, we know of no empirical results that back-up this intuition, and only of one paper (Haghighi and Klein, 2006) that actually evaluates an EM-based unsupervised constituent labeling algorithm (as a baseline for a much better semi-

[1] Instead, these algorithms use (i) a specialized, generative constituent-distituent model (Klein and Manning, 2002a), (ii) a generative dependency model (Klein and Manning, 2004), (iii) a non-generative exemplar-based model (Dennis, 2005), (iv) a generative tree-substitution model (Bod, 2006) and a non-generative dependency-like model (Seginer, 2007).

supervised model), but with disappointing results (see section 7). Even the issue of the evaluation of the categories is by no means trivial, and no standardized measure is agreed upon until now.

In this paper, we develop an unsupervised constituent labeling algorithm that outperforms the EM-based algorithm and has comparable performance to the supervised treebank PCFG on the development set. We base ourselves on the elegant framework proposed by (Stolcke, 1994; Stolcke and Omohundro, 1994), called Bayesian Model Merging (henceforth, BMM). Unlike this work, our algorithm takes bracketed sentences as input; like the full BMM, our algorithm proceeds by merging nonterminal labels to maximize a Bayesian objective function. The initial conditions, merge operation and objective function are described in section 2. In section 3 we give our optimizations that allow for empirical evaluation on large, benchmark corpora, and in sections 4 experimental results and an analysis of the model's strengths and weaknesses. Apart from evaluating the categories against a manually annotated gold standard, we are also interested in a theory-independent measure of performance. In section 4 we present results that compare the likelihood of the parses of the grammar induced by our labeling algorithm to the parses of the treebank grammar on a test set. Finally, in section 5 we describe a version of the algorithm that does complete unsupervised induction, involving both labeling and bracketing, but show that our objective function is not appropriate for this task.

2 Search and Objective Function

BMM defines a heuristic, greedy search for an optimal probabilistic context free grammar (PCFG) according to the Bayesian criterion of maximum a posteriori probability. In the present version, a single operator $\text{MERGE}(G, X_1, X_2) \mapsto G'$ defines possible transitions between grammars (Cook et al., 1976): it replaces non-terminal X_2 with non-terminal X_1 throughout grammar G, yielding a new grammar G'. MERGE creates generalizations by forming disjunctive groups (categories) of patterns that occur in the same contexts. In the search, all candidate merges are considered, and a single one is selected that most improves the objective function. The process is iterated until a state is reached where no single merge improves the objective function anymore. The search is augmented with a look-ahead procedure, that looks at a sequence of by default 10 subsequent merges.

The algorithm takes as input unlabeled sentences, with bracket information from the treebank (or from a specialized unsupervised bracketing algorithm). The initial rules of the grammar are read off from all productions implicit in the bracketed corpus, where every constituent, except for the start symbol, is given a unique label. The vast number of unique labels thus obtained is reduced to about half its size by merging, in a preprocessing step, labels of constituents which have exactly the same descendants. For example, the annotated sentence *(S (NP-SBJ (NNP Mr.) (NNP Ehrlich)) (VP (MD will) (VP (VB continue) (PP-CLR (IN as) (NP (NP (DT a) (NN director)) (CC and) (NP (DT a) (NN consultant)))))))*, is incorporated in the grammar as follows:

$$\begin{array}{lll|lll} S & \rightarrow X_0\ X_1 & (1) & X_3 & \rightarrow IN\ X_4 & (1) \\ X_0 & \rightarrow NNP\ NNP & (1) & X_4 & \rightarrow X_5\ CC\ X_5 & (1) \\ X_1 & \rightarrow MD\ X_2 & (1) & X_5 & \rightarrow DT\ NN & (2) \\ X_2 & \rightarrow VB\ X_3 & (1) & & & \end{array} \quad (1)$$

Possible merges are evaluated by the posterior probability of the resulting grammar. The maximum a posteriori (MAP) hypothesis, M_{MAP} is the hypothesis that maximizes the posterior probability. With Bayes Law: $M_{MAP} \equiv argmax_M P(M|X) = argmax_M P(X|M) \cdot P(M)$ where $P(X|M)$ is the likelihood of data X given grammar M, and $P(M)$ is the prior probability of the grammar. Maximizing $P(X|M) \cdot P(M)$ is equivalent to minimizing

$$-logP(M) - logP(X|M) \approx GDL + DDL = DL$$

This equation is interpreted in information theory as the total description length (DL): The Grammar Description Length $GDL = -logP(M)$ is the number of bits needed to encode the grammar (rounded to an integer number and assuming an optimal, shared code) and the Data Description Length $DDL = -logP(X|M)$ is code-length needed to describe the data given the model (Solomonoff, 1964). The MAP hypothesis is therefore the grammar with *Minimum Description Length*.

Structure Prior Using the description length interpretation allows for an intuitive way to choose the prior probability such that smaller grammars are favored[2]. We adopted the encoding scheme from (Petasis et al., 2004), which divides the grammar into top-productions (the set R_1), lexical productions (R_2) and other non-lexical productions (R_3). Rules from R_1 need one symbol less to encode than rules from R_3, because their LHS is fixed. N_r is the number of non-terminals in the RHS of a production r. Each non-terminal symbol requires $log(\mathcal{N}+1)$ bits to encode, where $\mathcal{N}$ is the number of unique nonterminals, and the 1 is for an end marker. T is the number of terminals ($|R_2| = T$), and 2 further end symbols are needed as separators of the three rule sets. Hence, the grammar description length is given by:

$$\begin{aligned} GDL &= \log(\mathcal{N}+1) \cdot \textstyle\sum_{r \in R_1} (N_r + 1) + (\log(\mathcal{N}+1) + \log(T)) \cdot T \\ &+ \log(\mathcal{N}+1) \cdot \textstyle\sum_{r \in R_3} (N_r + 2) + \log(\mathcal{N}+1) \cdot 2 \end{aligned} \quad (2)$$

Likelihood Assuming independence between the sentences, the likelihood of the corpus is the product of the likelihood of the sentences.

$$P(X|M) = \prod_{x \in X} \sum_{der:\ yield(der)=x} P(der|M) \quad (3)$$

The BMM algorithm makes two further approximations in the calculation of the likelihood (Stolcke and Omohundro, 1994). First, it is assumed that most of the probability mass of the sentence is concentrated in the Viterbi parse (the most probable derivation), so that the contribution of all non-Viterbi parses to the sentence probability and hence to the likelihood are ignored. Secondly, it is assumed that the merging operation preserves the Viterbi parse. This means that after a merge operation the Viterbi parse of the sentence is generated by exactly the same sequence of rewrite rules as before, except for the rules affected by the merge. We will come back to the validity of these approximations in section 5.

Using these approximations, we can compute the data likelihood directly from the grammar, if we keep track of the number of times that every rule is used in the entire corpus. This can be seen by rearranging the equation of likelihood, regrouping the rules used in all the samples according to their left hand side (Stolcke, 1994):

$$\begin{aligned} P(X|M) &= \textstyle\prod_{x \in X} \sum_{yield(der)=x} P(der) \approx \prod_{x \in X} P(der(X)_V) \\ &= \textstyle\prod_{x \in X} \prod_{r_i \in der_V} P(r_i)^{C_i} \prod_{A \in V_N} \prod_{r_i : A = lhs(r_i)} P(r_i)^{CC_i} \end{aligned} \quad (4)$$

where der is a derivation, der_V is the Viterbi parse, $r_i \in \mathcal{R}$ is a rewrite rule, $A \in V_N$ a nonterminal, C_i the count of rules occurring within a single derivation and CC_i the count of rules occurring in the Viterbi parses of the entire corpus.

3 Forecasting DL-gain

In our search for the grammar that maximizes the objective function, we will need to consider an enormous grammar space. At each time step, the number of alternative grammars reachable with one merge is quadratic in the number of nonterminals. It is therefore computationally intractable to calculate the posterior probability of each candidate grammar. Evaluating the BMM algorithm on realistically sized corpora therefore seemed infeasible (Klein, 2005; Clark, 2001). However, (Petasis et al., 2004) show that the DL-gain of chunks and merges can be efficiently predicted without having to consider a complete alternative grammar for every candidate search operation. Their equations can be adapted for the various choices of objective functions discussed above. The complexity of finding the best chunk is reduced from $O(\mathcal{N}^2)$ to $O(\mathcal{N})$, while the complexity of finding the best merge is reduced from $O(\mathcal{N}^3)$ to $O(\mathcal{N}^2)$ (Petasis et al., 2004).

After the application of a merge, the grammar is made smaller through elimination of duplicate rules: Ω_1 is the set of rules from R_1 that are eliminated, and Ω_3 is the set of rules from R_3 that are

[2]We are aware that there is much more to be said about the relation between Bayesian Inference and MDL, and that there might be much more linguistically motivated ways to choose priors (see e.g. (Eisner, 2002)). Here we take a pragmatic approach, however, aimed at defining simple priors that nevertheless force the algorithm to generalize beyond the training data.

eliminated. The change of the GDL as a result of the merge is (Petasis et al., 2004):

$$\begin{aligned}\Delta GDL_M &= \log\left(\frac{\mathcal{N}}{\mathcal{N}+1}\right) \times \left(\sum_{r \in R_1 \cap R_3} (N_r + 1) + T + |R_3| + 2\right) \\ &- \log(\mathcal{N}) \cdot \sum_{r \in \Omega_1 \cap \Omega_3} (N_r + 1) \end{aligned} \quad (5)$$

As before, the first term expresses the fact that the number of bits needed to encode any single non-terminal is changed due to the decrease in the total number of non-terminals, and this term is independent of the choice of the merge.

For the computation of the gain in data description length (DDL) as a result of merging the non-terminals X and Y, we can best view the merging process as two subprocesses:

- M^1 the rule sets with LHSs X and Y are joined (receive same LHS), changing the conditional probability of those rules and thus the DDL.
- M^2 duplicate rules that may occur as a result of the merge are eliminated in the entire grammar.

Although (Petasis et al., 2004) make this observation, their equations for ΔDDL cannot be used in our model, because they (implicitly) assume rule probabilities are uniformly distributed. That is, their E-GRIDS model assumes CFGs, whereas we work with PCFGs and, like (Stolcke, 1994), assume that rule probabilities are proportional to their frequencies in the Viterbi parses. This requires a modification of the formulas expressing the contribution of the merging operator to DDL.

From eq. 4, we see that the contribution of a non-terminal to the DDL is given by:

$$DDL_X = -\sum_{r:\ X=lhs(r)} F_r \cdot \big(log\left(\frac{F_r}{F_{Tot_X}}\right)\big)$$

where F_r is the frequency of a rule r with LHS X, and F_{Tot_X} is the sum of the frequencies of all rules with LHS X.

By joining the rules with LHS X and with LHS Y into a single set of rules, the relative frequency of a single rule is changed from $(\frac{F_r}{F_{Tot_X}})$ to $(\frac{F_r}{F_{Tot_{X+Y}}})$. This results in an overall gain in DDL of:

$$\Delta DDL_{M^1} = -\sum_{r:X=lhs(r)} \left(F_r \cdot \log(\frac{F_{Tot_X}}{F_{Tot_{X+Y}}})\right) - \sum_{r:X=lhs(r)} \left(F_r \cdot \log(\frac{F_{Tot_Y}}{F_{Tot_{X+Y}}})\right) \quad (6)$$

The gain in DDL from elimination of duplicate rules is given by:

$$\begin{aligned}\Delta DDL_{M^2} &= -\sum_{W \in \theta} \sum_{\omega \in \Omega_W} \Big[\sum_{r \in \omega} F_r \cdot \log(\sum_{k \in \omega} F_k / \sum_{l:lhs(l)=W} F_l) \\ &\quad -\sum_{r \in \omega} \big(F_r \cdot \log(F_r / \sum_{l:lhs(l)=W} F_l)\big)\Big] \end{aligned} \quad (7)$$

where θ is the set of LHS nonterminals of the duplicate rules resulting from merging X and Y. We thus sum over all LHS non-terminals W that have duplicate rules, and over all sets ω of duplicate rules. (For efficiency, our implementation maintains a datastructure that represents the DL gain of every pair of nonterminals that can be potentially merged, and updates this at every merge.)

4 Metrics and experimental evaluation

The BMM algorithm was trained and evaluated on the entire WSJ10 corpus. WSJ10 is the portion of 7422 sentences of length ≤ 10, after removal of punctuation and traces, extracted from the Penn Treebank Wall Street Journal (WSJ) (Marcus et al., 1993). It has been the prime benchmark used in recent grammar induction research. The POS-sequences are used as input for the induction algorithm, resulting in a vocabulary of 35 POS-tags. We left out sentences consisting of a single word

and sentences consisting of a repetition of the same word, to avoid spurious merges. Using the optimizations described above, the run on the 7422 sentences of the WSJ10 corpus lasted approximately 10 hours on a PC with 0.5 Gb memory.

Evaluating the quality of induced syntactic categories is difficult, and no widely agreed upon measure exists. We use two different metrics, related to those used in supervised parsing (labeled precision/recall) and speech recognition (likelihood/perplexity). For the first, the difficulty is that in unsupervised labeling algorithms, categories receive arbitrary internal labels; in order to evaluate them using labeled precision and recall, the induced labels must somehow be mapped onto the treebank labels. We follow (Haghighi and Klein, 2006), who use for their unsupervised baseline model a greedy remapping of the induced labels to the best matching tree bank label. This style of remapping, however, allows for multiple induced labels to map to a single target label. With a fixed number of categories that is no major problem, but with many more induced non-terminals the measure is too optimistic: in the extreme case of a unique induced label for every constituent it would give 100% precision and recall. Since in most of our experiments the number of experimental labels exceeded the number of treebank labels by a large number, we measured labeled recall by defining the remapping the other way round, from the treebank labels to the induced labels. That is, we replace each tree bank label with its best matching induced label, and measure recall with this transformed treebank as gold standard. The F-score is still defined as the harmonic mean of LP and LR: $F_1 = \frac{2*LP*LR}{LP+LR}$.

The motivation for this way of calculating precision and recall is somewhat involved. Consider that syntactic categories are meant to be defining substitutability. Every label X that is used for N_X constituents in the induced trees, thus defines $N_X \cdot (N_X - 1)$ substitutions that are grammatical according to the induced grammar G_i. If out of these N_X constituents, $M_{X,Y}$ constituents receive the same label Y in the gold-standard treebank, that means that at least $M_{X,Y} \cdot (M_{X,Y} - 1)$ of the substitutions that are grammatical according to G_i are also grammatical according to the gold-standard grammar G_g. Although there might be other categories Y' that permit other substitutions, the portion of the $N_X \times (N_X - 1)$ substitutions that is also permitted by G_g will be dominated by $Z_{max} = \text{argmax}_Z M_{X,Z}$. Hence, $M_{Z_{max}}$ is a lower bound and a good approximation of the square root of the number of X-substitutions permitted by G_g ("substitutability precision"). Similar reasoning on the number of treebank substitution permitted by G_i ("substitutability recall") leads to the measures proposed (ignoring for the moment some issues with averaging results).

A drawback of this style of evaluating, is that it is still completely dependent on the manual treebank annotation, which is far from theory-neutral. In fact, all state-of-the-art supervised parsing and language models, can be viewed as redefining the treebank nonterminal labels. Our second style of evaluation avoids this dependency. It is based on splitting the corpus in a train and test set, and measuring the likelihood (or perplexity) of the test set according to the induced grammar. Grammars that overfit the trainset, or generalize too much to unlikely or ungrammatical sentences, will give low likelihood to test set sentences. For these experiments we use the traditional sections 2-21 of WSJ10 as trainset, and section 22 as test set (section 23 is kept for future evaluations).

In table 1(a) we report our relabeled recall and precision scores. For comparison, we also give the baseline results (Haghighi and Klein, 2006) obtained by running the Inside-Outside algorithm on grammar initialized with all binary rules that can be built from the treebank syntactic categories. We also copy their results with a semi-supervised "proto-type" driven induction algorithm run on the same data. Our unsupervised algorithm obtained a (relabeled) F-score of 76.8, which compares favorably to the (labeled) F-score of 71.1 from that paper. Note however, that different definitions of LP and LR are used. In table 1(b) we give the relabeled precision scores of the 7 most frequent categories (after relabeling) in our induced trees. For most frequent categories (leaving out the S category) precision is near or above 90% except for the NP category.

For a second set of experiments, the algorithm was trained on sections 2-21 of WSJ10, and evaluated on section 22. Figure 1(a) gives the sum of the likelihood of all testset sentences, as parsed by the baseline "unlabeled" (the treebank PCFG after all nonterminals have been replaced by an X), the treebank PCFG (labeled TBG) and the grammar that results from our BMM-algorithm (the latter grammar failed to parse 6 sentences up to length 20, and 4 up to length 10. These were excluded from all evaluations.) Figure 1(b) shows that the vast majority of the sentences in the test set receives higher likelihood from the grammar induced by BMM than from either the treebank PCFG or the unlabeled grammar. Note that the grammar that uses a unique nonterminal for every constituent in

(a)

Model	LP	LR	F
In-Out	47.0	57.2	51.6
Proto	64.8	78.7	71.1
BMM (all)	75.1	78.5	76.8

(b)

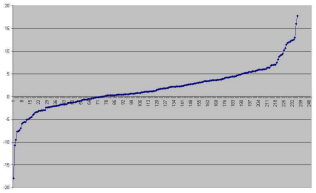
Label	LP	Freq
NP	57.8	38.5%
VP	92.0	22.3%
PP	89.0	8.1%
ADVP	100.0	4.0%
ADJP	100.0	2.7%
QP	89.3	1.6%
SBAR	100.0	1.4%

Table 1: (a) Results of several algorithms on the unsupervised labeling task (using gold standard bracketing). The results with the unsupervised Inside-Outside algorithm and the semi-supervised Prototype-Driven Grammar Induction-algorithm (labeled Proto) are from (Haghighi and Klein, 2006). (b) Relabeled precision scores per category (after mapping to the treebank categories).

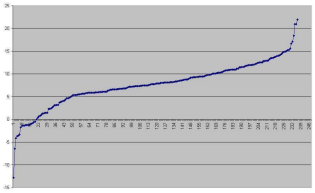
the treebank, such as BMM uses as initial condition, would score even worse: it assigns a likelihood of 0 to all unseen sentences (i.e. -log likelihood of ∞ to the testset).

G	$s \in$ WSJ10	WSJ20
TBG	7196	47266
BMM	6783	46612
unlab	8592	55961

(a) $\sum_s -\log P(s|G)$



(b) TBG-BMM



(c) unlabeled-BMM

Figure 1: (a) The summed -log likelihood of sentences from section 22, (b) Log likelihood differences between TBG and BMM on all sentences of WSJ10 (y-axis gives the P(sen—G), x-axis gives rank when sentences are ordered on y-axis value), (c) Log likelihood differences between the unlabeled grammar and BMM.

5 Combining Unsupervised Labeling with Unsupervised Bracketing

We have also tried to use the BMM algorithm for the task of simultaneous bracketing and labeling. For this purpose, we add the chunk operator: $\text{CHUNK}(G, X_1, X_2) \mapsto G'$, which takes a sequence of two nonterminals X_1 and X_2 and creates a unique new nonterminal Y that expands to $X_1 X_2$. The merging and chunking operations now proceed in two alternating phases: in the merging phase, the search is iterated until a state is reached where no single merge improves the objective function anymore. In the chunking phase only one best chunk is selected. As in (Stolcke and Omohundro, 1994), the initial rules of the grammar are set to incorporate the complete samples, and the bracketing information from the treebank is discarded.

By adding the chunk operation, we have effectively recovered the full Bayesian Model Merging algorithm of (Stolcke and Omohundro, 1994). Because of our optimizations, we are able to perform the first evaluation of that algorithm on the benchmark test WSJ10. Tables 2 summarize results from experiments with the BMM algorithm on WSJ10 (our best results use the "Poisson prior" with $\mu = 3$; for details see (Stolcke, 1994)). Unfortunately, the completely unsupervised BMM algorithm does not meet up to the standards of previous work on unsupervised bracketing and even to the right branching baseline (R-B).

We investigated a number of possible explanations for the poor performance. First, we considered the possibility that our assumption that Viterbi parses are mostly preserved after the application of chunks and merges (section 2) does not hold. We parsed the entire corpus again with the induced grammar, and compared the resulting parses with the parses assumed in the approximation. The differences were relatively minor when measured with the same metric as used for comparison

	WSJ-10	UP	UR	F
	R-B	70.0	55.1	61.7
	CCM (Klein and Manning, 2002b)	64.2	81.6	71.9
	DMV+CCM (Klein and Manning, 2004)	69.3	88.0	77.6
	U-DOP (Bod, 2006)	70.8	88.2	78.5
	E-GRIDS (Petasis et al., 2004)	59.3	37.8	46.2
(a)	**BMM (this paper)**	**57.6**	**43.1**	**49.3**

	($\times 10^6$)	DL	GDL	DDL
	Init	.295	.066	.226
	Final	.276	.041	.235
		UP	**UR**	**F**
	Init	90.1	88.6	89.4
(b)	**Final**	64.3	74.8	69.2

Table 2: (a) Results of BMM compared to previous work on the same data; (b) BMM initialized with the treebank grammar

against gold-standard parses, with $UP = UR = 95.9$ on OVIS, and $UP = 95.1$ and $UP = 94.2$ on WSJ10. Hence, our approximations seem to be justified. A second possibility is that the heuristic search procedure used is unable to find the high-accuracy regions of the search space. To test this, we used the treebank PCFG as the initial grammar of the BMM algorithm, which yields comparatively very high UP and UR scores. As table 2(b) shows, this treebank grammar is not an optimum of the objective function used: the BMM algorithm continues for a long time to improve the description length, whilst the F-score against the gold standard parses monotonously decreases.

A qualitative analysis of the merges and chunks in the process further shows a number of problems. First, there are many ungrammatical chunks, which are formed by cutting across constituent boundaries, e.g. $put\ the$, $on\ the$, $and\ a$, $up\ and$. This is explained by the fact that the prime criterion for selecting a chunk is the bigram frequency. Second, there is a tendency for categories to overgeneralize. This effect seems to be self-reinforcing. Merging errors are carried over to the chunking phase and vice versa, causing a snow ball effect. Eventually, most categories cluster together. These experiments indicate that it is not a failure of the approximations or the search algorithm which prevents the algorithm from reaching the optimal grammar, but rather a wrong choice of the objective function. Better choices for the prior probability distribution are a major topic for future research.

6 Discussion and Conclusions

In this paper we have studied the problem of labeling constituents in a bracketed corpus. We have argued that this is an important task, complementing the task of unsupervised bracketing on which much progress has been made in recent years (Klein and Manning, 2002b; Bod, 2006). Together, unsupervised bracketing and unsupervised labeling hold the promise of (i) accounting for the unsupervised acquisition of grammar by children, and (ii) relieving research in (multilingual) NLP of its sparseness of annotated data. Our algorithm uses the PCFG formalism and starts out with a grammar that models the training data perfectly, but does not generalize beyond it. It is important to note that the right sequence of merges can take us to any PCFG consistent with the bracketing. Although natural language contains some constructions that are difficult or, in exceptional cases, impossible to model with PCFGs, the formalism is rich enough to encode extremely accurate grammars if (and only if) the traditional linguistic categories are dropped (see e.g. (Petrov et al., 2006))[3]. The key question is thus how to guide the model merging algorithm through the space of possible PCFGs.

[3]For smoothing, (Petrov et al., 2006) use information internal to non-terminal labels, which takes their model outside the class of PCFGs. However, without smoothing they already obtain surprisingly good results.

We have used a Bayesian objective function, in the tradition of (Solomonoff, 1964), to guide this search, and implemented a number of heuristics to perform this search efficiently, using techniques from (Cook et al., 1976), (Stolcke and Omohundro, 1994) and (Petasis et al., 2004). We have found that our BMM-algorithm performs better than state-of-the-art unsupervised labeling algorithms on (re)labeled precision and recall metrics. We have further shown that the likelihood our induced grammars assign to unseen sentences, rivals that of the treebank PCFG (although it presumably performs worse than state-of-the-art supervised language models). These positive results suggest it is possible to devise hybrid models, where specialized bracketers such as (Klein and Manning, 2004) and (Bod, 2006), can be combined with BMM as a specialized unsupervised labeling algorithm.

In section 5, we have demonstrated briefly that, contrary to received wisdom (Klein and Manning, 2005), (Clark, 2001) the full BMM framework of (Stolcke and Omohundro, 1994), which includes bracketing, can be evaluated on large corpora. We think this is an important step in its own right. There is a rich body of research from previous decades on unsupervised grammar induction. We have shown that it can be worthwhile to use an older algorithm, and evaluate it according to current experimental methodology. However, the performance of completely unsupervised BMM on real languages is rather disappointing. The fact that BMM can still optimize the description length when initialized with the treebank grammar indicates that the problem is probably not with the search, but with the applicability of the objective function for natural languages. The distinction BMM makes between prior, likelihood and heuristic search, allows us to now focus our attention on the prior, without having to replace the entire technical apparatus we and others developed.

The finding that the same objective function is appropriate for merging, but not at all for chunking, is interesting in its own right. Because the structure of natural language reflects the learning biases of language learners (Kirby et al., 2007; Zuidema, 2003), this finding might be relevant for theories of language acquisition. Perhaps children use distributional information, such as exploited by our algorithm, to a large extent for discovering syntactic categories, while relying on other information (semantic integrity, phonological phrasing) for identifying constituents in the first place. The finding that induced syntactic categories outperform traditional linguistic categories is in concordance with theories in cognitive linguistics that deny universal status of these categories, and view them as derived from specific linguistic constructions rather than as a-priori given (Croft, 2001).

Acknowledgements WZ is funded by the Netherlands Organisation for Scientific Research (EW), project nr. 612.066.405; many thanks to Yoav Seginer, Remko Scha and Rens Bod for their comments.

References

Angluin, D. and Smith, C. H. (1983). Inductive inference: Theory and methods. *Computing Surveys*, 15(3).

Bod, R. (2006). An all-subtrees approach to unsupervised parsing. *Proceedings ACL-COLING'06*.

Clark, A. (2001). *Unsupervised Language Acquisition: Theory and Practice*. PhD thesis, University of Sussex.

Cook, C., Rosenfeld, A., and Aronson, A. (1976). Grammatical inference by hill climbing. *Informational Sciences (now: Information Sciences)*, 10:59–80.

Croft, W. (2001). *Radical construction grammar: syntactic theory in typological perspective*. Oxford University Press, Oxford.

Dennis, S. (2005). An exemplar-based approach to unsupervised parsing. In Bara, B. G., Barsalou, L., and Bucciarelli, M., editors, *Proceedings of the 27th Conference of the Cognitive Science Society*. Lawrence Erlbaum.

Eisner, J. (2002). Transformational priors over grammars. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 63–70, Philadelphia. Association for Computational Linguistics.

Haghighi, A. and Klein, D. (2006). Prototype-driven grammar induction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*.

Hemphill, C., Godfrey, J., and Doddington, G. (1990). The ATIS spoken language systems pilot corpus. In *Proceedings of the DARPA Speech and Natural Language Workshop*. Morgan Kaufman, Hidden Valley.

Kirby, S., Dowman, M., and Griffiths, T. L. (2007). Innateness and culture in the evolution of language. *PNAS*, 104(12):5241–5245.

Klein, D. (2005). *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University.

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

Klein, D. and Manning, C. D. (2002b). Natural language grammar induction using a constituent-context model. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA. MIT Press.

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

Klein, D. and Manning, C. D. (2005). Natural language grammar induction with a generative constituent-context model. *Pattern Recognition*, 38.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2).

Petasis, G., Paliouras, G., Karkaletsis, V., Halatsis, C., and Spyropoulos, C. (2004). E-grids: Computationally efficient grammatical inference from positive examples. *Grammars*, 7:69–110.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In *Proceedings ACL-COLING'06*, pages 443–440. Association for Computational Linguistics Morristown, NJ, USA.

Pinker, S. (1979). Formal models of language learning. *Cognition*, 7:217–283.

Sakakibara, Y. (1997). Recent advances of grammatical inference. *Theoretical Computer Science*, 185:15–45.

Seginer, Y. (2007). Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384–391, Prague, Czech Republic. Association for Computational Linguistics.

Solomonoff, R. (1964). A formal theory of inductive inference, part ii. *Information and Control*, 7(2):224–254.

Stolcke, A. (1994). *Bayesian Learning of Probabilistic Language Models*. PhD thesis, Dept. of Electrical Engineering and Computer Science, University of California at Berkeley.

Stolcke, A. and Omohundro, S. M. (1994). Inducing probabilistic grammars by Bayesian model merging. In *Proc. Second International Colloquium on Grammatical Inference and Applications (ICGI'94)*, volume 862 of *Lecture Notes in Computer Science*, pages 106–118, Berlin. Springer-Verlag.

Zuidema, W. (2003). How the poverty of the stimulus solves the poverty of the stimulus. In Becker, S., Thrun, S., and Obermayer, K., editors, *Advances in Neural Information Processing Systems 15 (Proceedings of NIPS'02)*, pages 51–58. MIT Press, Cambridge, MA.

An All-Subtrees Approach to Unsupervised Parsing

Rens Bod
School of Computer Science
University of St Andrews
North Haugh, St Andrews
KY16 9SX Scotland, UK
rb@dcs.st-and.ac.uk

Abstract

We investigate generalizations of the all-subtrees "DOP" approach to unsupervised parsing. Unsupervised DOP models assign all possible binary trees to a set of sentences and next use (a large random subset of) all subtrees from these binary trees to compute the most probable parse trees. We will test both a relative frequency estimator for unsupervised DOP and a maximum likelihood estimator which is known to be statistically consistent. We report state-of-the-art results on English (WSJ), German (NEGRA) and Chinese (CTB) data. To the best of our knowledge this is the first paper which tests a maximum likelihood estimator for DOP on the Wall Street Journal, leading to the surprising result that *an unsupervised parsing model beats a widely used supervised model (a treebank PCFG).*

1 Introduction

The problem of bootstrapping syntactic structure from unlabeled data has regained considerable interest. While supervised parsers suffer from shortage of hand-annotated data, unsupervised parsers operate with unlabeled raw data of which unlimited quantities are available. During the last few years there has been steady progress in the field. Where van Zaanen (2000) achieved 39.2% unlabeled f-score on ATIS word strings, Clark (2001) reports 42.0% on the same data, and Klein and Manning (2002) obtain 51.2% f-score on ATIS part-of-speech strings using a constituent-context model called CCM. On Penn Wall Street Journal p-o-s-strings ≤ 10 (WSJ10), Klein and Manning (2002) report 71.1% unlabeled f-score with CCM. And the hybrid approach of Klein and Manning (2004), which combines constituency and dependency models, yields 77.6% f-score.

Bod (2006) shows that a further improvement on the WSJ10 can be achieved by an unsupervised generalization of the all-subtrees approach known as Data-Oriented Parsing (DOP). This unsupervised DOP model, coined U-DOP, first assigns all possible unlabeled binary trees to a set of sentences and next uses all subtrees from (a large subset of) these trees to compute the most probable parse trees. Bod (2006) reports that U-DOP not only outperforms previous unsupervised parsers but that its performance is as good as a binarized *supervised* parser (i.e. a treebank PCFG) on the WSJ.

A possible drawback of U-DOP, however, is the statistical inconsistency of its estimator (Johnson 2002) which is inherited from the DOP1 model (Bod 1998). That is, even with unlimited training data, U-DOP's estimator is not guaranteed to converge to the correct weight distribution. Johnson (2002: 76) argues in favor of a maximum likelihood estimator for DOP which *is* statistically consistent. As it happens, in Bod (2000) we already developed such a DOP model, termed ML-DOP, which reestimates the subtree probabilities by a maximum likelihood procedure based on Expectation-Maximization. Although cross-validation is needed to avoid overlearning, ML-DOP outperforms DOP1 on the OVIS corpus (Bod 2000).

This raises the question whether we can create an *unsupervised* DOP model which is also

statistically consistent. In this paper we will show that an unsupervised version of ML-DOP can be constructed along the lines of U-DOP. We will start out by summarizing DOP, U-DOP and ML-DOP, and next create a new unsupervised model called UML-DOP. We report that UML-DOP not only obtains higher parse accuracy than U-DOP on three different domains, but that it also achieves this with *fewer* subtrees than U-DOP. To the best of our knowledge, this paper presents the first *unsupervised* parser that outperforms a widely used *supervised* parser on the WSJ, i.e. a treebank PCFG. We will raise the question whether the end of supervised parsing is in sight.

2 DOP

The key idea of DOP is this: given an annotated corpus, use all subtrees, regardless of size, to parse new sentences. The DOP1 model in Bod (1998) computes the probabilities of parse trees and sentences from the relative frequencies of the subtrees. Although it is now known that DOP1's relative frequency estimator is statistically inconsistent (Johnson 2002), the model yields excellent empirical results and has been used in state-of-the-art systems. Let's illustrate DOP1 with a simple example. Assume a corpus consisting of only two trees, as given in figure 1.

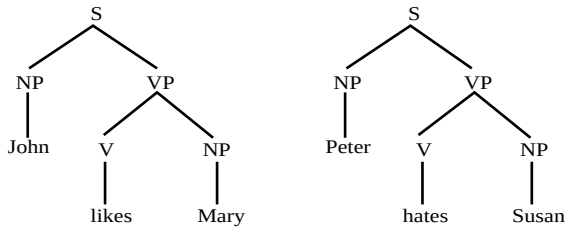


Figure 1. A corpus of two trees

New sentences may be derived by combining fragments, i.e. subtrees, from this corpus, by means of a node-substitution operation indicated as ∘. Node-substitution identifies the leftmost nonterminal frontier node of one subtree with the root node of a second subtree (i.e., the second subtree is *substituted* on the leftmost nonterminal frontier node of the first subtree). Thus a new sentence such as *Mary likes Susan* can be derived by combining subtrees from this corpus, shown in figure 2.

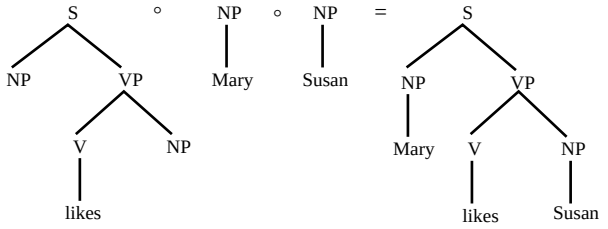


Figure 2. A derivation for *Mary likes Susan*

Other derivations may yield the same tree, e.g.:

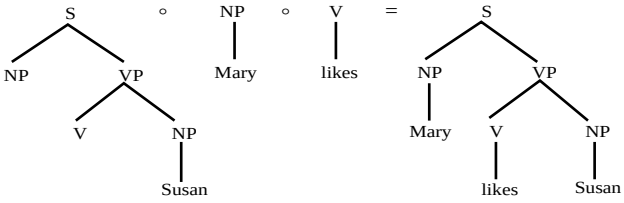


Figure 3. Another derivation yielding same tree

DOP1 computes the probability of a subtree t as the probability of selecting t among all corpus subtrees that can be substituted on the same node as t. This probability is computed as the number of occurrences of t in the corpus, $|\, t \,|$, divided by the total number of occurrences of all subtrees t' with the same root label as t.[1] Let $r(t)$ return the root label of t. Then we may write:

$$P(t) = \frac{|\, t \,|}{\sum_{t':\, r(t')=r(t)} |\, t' \,|}$$

The probability of a derivation $t_1 \circ ... \circ t_n$ is computed by the product of the probabilities of its subtrees t_i:

$$P(t_1 \circ ... \circ t_n) = \prod_i P(t_i)$$

As we have seen, there may be several distinct derivations that generate the same parse tree. The probability of a parse tree T is the sum of the

[1] This subtree probability is redressed by a simple correction factor discussed in Goodman (2003: 136) and Bod (2003).

probabilities of its distinct derivations. Let t_{id} be the i-th subtree in the derivation d that produces tree T, then the probability of T is given by

$$P(T) = \sum_d \prod_i P(t_{id})$$

Thus DOP1 considers counts of subtrees of a wide range of sizes: everything from counts of single-level rules to entire trees is taken into account to compute the most probable parse tree of a sentence. A disadvantage of the approach may be that an extremely large number of subtrees (and derivations) must be considered. Fortunately there exists a compact isomorphic PCFG-reduction of DOP1 whose size is linear rather than exponential in the size of the training set (Goodman 2003). Moreover, Collins and Duffy (2002) show how a tree kernel can be applied to DOP1's all-subtrees representation. The currently most successful version of DOP1 uses a PCFG-reduction of the model with an n-best parsing algorithm (Bod 2003).

3 U-DOP

U-DOP extends DOP1 to unsupervised parsing (Bod 2006). Its key idea is to assign all unlabeled binary trees to a set of sentences and to next use (in principle) all subtrees from these binary trees to parse new sentences. U-DOP thus proposes one of the richest possible models in bootstrapping trees. Previous models like Klein and Manning's (2002, 2005) CCM model limit the dependencies to "contiguous subsequences of a sentence". This means that CCM neglects dependencies that are *non*-contiguous such as between *more* and *than* in "*BA carried more people than cargo*". Instead, U-DOP's all-subtrees approach captures both contiguous and non-contiguous lexical dependencies.

As with most other unsupervised parsing models, U-DOP induces trees for p-o-s strings rather than for word strings. The extension to word strings is straightforward as there exist highly accurate unsupervised part-of-speech taggers (e.g. Schütze 1995) which can be directly combined with unsupervised parsers.

To give an illustration of U-DOP, consider the WSJ p-o-s string NNS VBD JJ NNS which may correspond for instance to the sentence *Investors suffered heavy losses*. U-DOP starts by assigning all possible binary trees to this string, where each root node is labeled *S* and each internal node is labeled *X*. Thus NNS VBD JJ NNS has a total of five binary trees shown in figure 4 -- where for readability we add words as well.

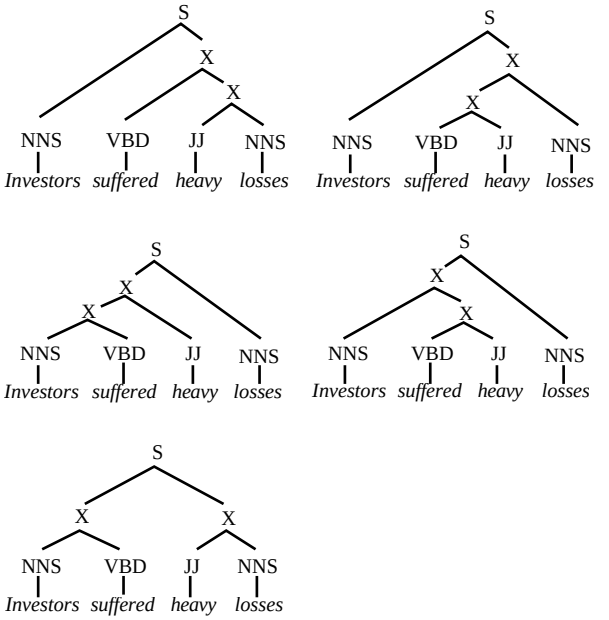


Figure 4. All binary trees for NNS VBD JJ NNS (*Investors suffered heavy losses*)

While we can efficiently represent the set of all binary trees of a string by means of a chart, we need to unpack the chart if we want to extract subtrees from this set of binary trees. And since the total number of binary trees for the small WSJ10 is almost 12 million, it is doubtful whether we can apply the unrestricted U-DOP model to such a corpus. U-DOP therefore randomly samples a large subset from the total number of parse trees from the chart (see Bod 2006) and next converts the subtrees from these parse trees into a PCFG-reduction (Goodman 2003). Since the computation of the most probable parse tree is NP-complete (Sima'an 1996), U-DOP estimates the most probable tree from the 100 most probable derivations using Viterbi n-best parsing. We could also have used the more efficient k-best hypergraph parsing technique by Huang and Chiang (2005), but we have not yet incorporated this into our implementation.

To give an example of the dependencies that U-DOP can take into account, consider the following subtrees in figure 5 from the trees in

figure 4 (where we again add words for readability). These subtrees show that U-DOP takes into account both contiguous and non-contiguous substrings.

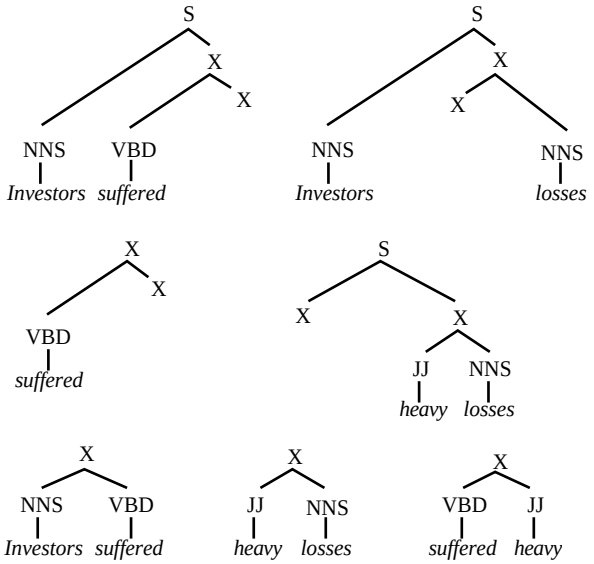


Figure 5. Some subtrees from trees in figure 4

Of course, if we only had the sentence *Investors suffered heavy losses* in our corpus, there would be no difference in probability between the five parse trees in figure 4. However, if we also have a different sentence where JJ NNS (*heavy losses*) appears in a different context, e.g. in *Heavy losses were reported*, its covering subtree gets a relatively higher frequency and the parse tree where *heavy losses* occurs as a constituent gets a higher total probability.

4 ML-DOP

ML-DOP (Bod 2000) extends DOP with a maximum likelihood reestimation technique based on the expectation-maximization (EM) algorithm (Dempster et al. 1977) which is known to be statistically consistent (Shao 1999). ML-DOP reestimates DOP's subtree probabilities in an iterative way until the changes become negligible. The following exposition of ML-DOP is heavily based on previous work by Bod (2000) and Magerman (1993).

It is important to realize that there is an implicit assumption in DOP that all possible derivations of a parse tree contribute equally to the total probability of the parse tree. This is equivalent to saying that there is a hidden component to the model, and that DOP can be trained using an EM algorithm to determine the maximum likelihood estimate for the training data. The EM algorithm for this ML-DOP model is related to the Inside-Outside algorithm for context-free grammars, but the reestimation formula is complicated by the presence of subtrees of depth greater than 1. To derive the reestimation formula, it is useful to consider the state space of all possible derivations of a tree.

The derivations of a parse tree T can be viewed as a state trellis, where each state contains a partially constructed tree in the course of a leftmost derivation of T. s_t denotes a state containing the tree t which is a subtree of T. The state trellis is defined as follows.

The initial state, s_0, is a tree with depth zero, consisting of simply a root node labeled with S. The final state, s_T, is the given parse tree T.

A state s_t is connected forward to all states s_{tf} such that $t_f = t \circ t'$, for some t'. Here the appropriate t' is defined to be $t_f - t$.

A state s_t is connected backward to all states s_{tb} such that $t = t_b \circ t'$, for some t'. Again, t' is defined to be $t - t_b$.

The construction of the state lattice and assignment of transition probabilities according to the ML-DOP model is called the forward pass. The probability of a given state, $P(s)$, is referred to as $\alpha(s)$. The forward probability of a state s_t is computed recursively

$$\alpha(s_t) = \sum_{s_{t_b}} \alpha(s_{t_b})\, P(t - t_b).$$

The backward probability of a state, referred to as $\beta(s)$, is calculated according to the following recursive formula:

$$\beta(s_t) = \sum_{s_{t_f}} \beta(s_{t_f})\, P(t_f - t)$$

where the backward probability of the goal state is set equal to the forward probability of the goal state, $\beta(s_T) = \alpha(s_T)$.

The update formula for the count of a subtree t is (where $r(t)$ is the root label of t):

$$ct(t) = \sum_{s_{t_b}: \exists s_{t_f}, t_b \circ t = t_f} \frac{\beta(s_{t_f})\alpha(s_{t_b})P(t \mid r(t))}{\alpha(s_{goal})}$$

The updated probability distribution, $P'(t \mid r(t))$, is defined to be

$$P'(t \mid r(t)) = \frac{ct(t)}{ct(r(t))}$$

where $ct(r(t))$ is defined as

$$ct(r(t)) = \sum_{t':\ r(t')=r(t)} ct(t')$$

In practice, ML-DOP starts out by assigning the same relative frequencies to the subtrees as DOP1, which are next reestimated by the formulas above. We may in principle start out with any initial parameters, including random initializations, but since ML estimation is known to be very sensitive to the initialization of the parameters, it is convenient to start with parameters that are known to perform well.

To avoid overtraining, ML-DOP uses the subtrees from one half of the training set to be trained on the other half, and vice versa. This cross-training is important since otherwise UML-DOP would assign the training set trees their empirical frequencies and assign zero weight to all other subtrees (cf. Prescher et al. 2004). The updated probabilities are iteratively reestimated until the decrease in cross-entropy becomes negligible. Unfortunately, no compact PCFG-reduction of ML-DOP is known. As a consequence, parsing with ML-DOP is very costly and the model has hitherto never been tested on corpora larger than OVIS (Bonnema et al. 1997). Yet, we will show that by clever pruning we can extend our experiments not only to the WSJ, but also to the German NEGRA and the Chinese CTB. (Zollmann and Sima'an 2005 propose a different consistent estimator for DOP, which we cannot go into here).

5 UML-DOP

Analogous to U-DOP, UML-DOP is an unsupervised generalization of ML-DOP: it first assigns all unlabeled binary trees to a set of sentences and next extracts a large (random) set of subtrees from this tree set. It then reestimates the initial probabilities of these subtrees by ML-DOP on the sentences from a held-out part of the tree set. The training is carried out by dividing the tree set into two equal parts, and reestimating the subtrees from one part on the other. As initial probabilities we use the subtrees' relative frequencies as described in section 2 (smoothed by Good-Turing -- see Bod 1998), though it would also be interesting to see how the model works with other initial parameters, in particular with the usage frequencies proposed by Zuidema (2006).

As with U-DOP, the total number of subtrees that can be extracted from the binary tree set is too large to be fully taken into account. Together with the high computational cost of reestimation we propose even more drastic pruning than we did in Bod (2006) for U-DOP. That is, while for sentences ≤ 7 words we use all binary trees, for each sentence ≥ 8 words we randomly sample a fixed number of 128 trees (which effectively favors more frequent trees). The resulting set of trees is referred to as the binary tree set. Next, we randomly extract for each subtree-depth a fixed number of subtrees, where the depth of subtree is the longest path from root to any leaf. This has roughly the same effect as the correction factor used in Bod (2003, 2006). That is, for each particular depth we sample subtrees by first randomly selecting a node in a random tree from the binary tree set after which we select random expansions from that node until a subtree of the particular depth is obtained. For our experiments in section 6, we repeated this procedure 200,000 times for each depth. The resulting subtrees are then given to ML-DOP's reestimation procedure. Finally, the reestimated subtrees are used to compute the most probable parse trees for all sentences using Viterbi *n*-best, as described in section 3, where the most probable parse is estimated from the 100 most probable derivations.

A potential criticism of (U)ML-DOP is that since we use DOP1's relative frequencies as initial parameters, ML-DOP may only find a local maximum nearest to DOP1's estimator. But this is of course a criticism against any iterative ML approach: it is not guaranteed that the global maximum is found (cf. Manning and Schütze 1999: 401). Nevertheless we will see that our reestimation

procedure leads to significantly better accuracy compared to U-DOP (the latter would be equal to UML-DOP under 0 iterations). Moreover, in contrast to U-DOP, UML-DOP *can* be theoretically motivated: it maximizes the likelihood of the data using the statistically consistent EM algorithm.

6 Experiments: Can we beat supervised by unsupervised parsing?

To compare UML-DOP to U-DOP, we started out with the WSJ10 corpus, which contains 7422 sentences ≤ 10 words after removing empty elements and punctuation. We used the same evaluation metrics for unlabeled precision (UP) and unlabeled recall (UR) as defined in Klein (2005: 21-22). Klein's definitions differ slightly from the standard PARSEVAL metrics: multiplicity of brackets is ignored, brackets of span one are ignored and the bracket labels are ignored. The two metrics of UP and UR are combined by the unlabeled f-score F1 which is defined as the harmonic mean of UP and UR: F1 = 2*UP*UR/(UP+UR).

For the WSJ10, we obtained a binary tree set of $5.68 * 10^5$ trees, by extracting the binary trees as described in section 5. From this binary tree set we sampled 200,000 subtrees for each subtree-depth. This resulted in a total set of roughly $1.7 * 10^6$ subtrees that were reestimated by our maximum-likelihood procedure. The decrease in cross-entropy became negligible after 14 iterations (for both halfs of WSJ10). After computing the most probable parse trees, UML-DOP achieved an f-score of 82.9% which is a 20.5% error reduction compared to U-DOP's f-score of 78.5% on the same data (Bod 2006).

We next tested UML-DOP on two additional domains which were also used in Klein and Manning (2004) and Bod (2006): the German NEGRA10 (Skut et al. 1997) and the Chinese CTB10 (Xue et al. 2002) both containing 2200+ sentences ≤ 10 words after removing punctuation. Table 1 shows the results of UML-DOP compared to U-DOP, the CCM model by Klein and Manning (2002), the DMV dependency learning model by Klein and Manning (2004) as well as their combined model DMV+CCM.

Table 1 shows that UML-DOP scores better than U-DOP and Klein and Manning's models in all cases. It thus pays off to not only use subtrees rather than substrings (as in CCM) but to also reestimate the subtrees' probabilities by a maximum-likelihood procedure rather than using their (smoothed) relative frequencies (as in U-DOP). Note that UML-DOP achieves these improved results with *fewer* subtrees than U-DOP, due to UML-DOP's more drastic pruning of subtrees. It is also noteworthy that UML-DOP, like U-DOP, does not employ a separate class for non-constituents, so-called distituents, while CCM and CCM+DMV do. (Interestingly, the top 10 most frequently learned constituents by UML-DOP were exactly the same as by U-DOP -- see the relevant table in Bod 2006).

Model	English (WSJ10)	German (NEGRA10)	Chinese (CTB10)
CCM	71.9	61.6	45.0
DMV	52.1	49.5	46.7
DMV+CCM	77.6	63.9	43.3
U-DOP	78.5	65.4	46.6
UML-DOP	**82.9**	**67.0**	**47.2**

Table 1. F-scores of UML-DOP compared to previous models on the same data

We were also interested in testing UML-DOP on longer sentences. We therefore included all WSJ sentences up to 40 words after removing empty elements and punctuation (WSJ40) and again sampled 200,000 subtrees for each depth, using the same method as before. Furthermore, we compared UML-DOP against a supervised binarized PCFG, i.e. a treebank PCFG whose simple relative frequency estimator corresponds to maximum likelihood (Chi and Geman 1998), and which we shall refer to as "ML-PCFG". To this end, we used a random 90%/10% division of WSJ40 into a training set and a test set. The ML-PCFG had thus access to the Penn WSJ trees in the training set, while UML-DOP had to bootstrap all structure from the *flat* strings from the training set to next parse the 10% test set -- clearly a much more challenging task. Table 2 gives the results in terms of f-scores.

The table shows that UML-DOP scores better than U-DOP, also for WSJ40. Our results on WSJ10 are somewhat lower than in table 1 due to the use of a smaller training set of 90% of the data. But the most surprising result is that UML-DOP's f-

score is higher than the *supervised* binarized treebank PCFG (ML-PCFG) for both WSJ10 and WSJ40. In order to check whether this difference is statistically significant, we additionally tested on 10 different 90/10 divisions of the WSJ40 (which were the same splits as in Bod 2006). For these splits, UML-DOP achieved an average f-score of 66.9%, while ML-PCFG obtained an average f-score of 64.7%. The difference in accuracy between UML-DOP and ML-PCFG was statistically significant according to paired *t*-testing ($p \leq 0.05$). To the best of our knowledge this means that we have shown for the first time that *an unsupervised parsing model (UML-DOP) outperforms a widely used supervised parsing model (a treebank PCFG) on the WSJ40*.

Model	WSJ10	WSJ40
U-DOP	78.1	63.9
UML-DOP	**82.5**	**66.4**
ML-PCFG	81.5	64.6

Table 2. F-scores of U-DOP, UML-DOP and a supervised treebank PCFG (ML-PCFG) for a random 90/10 split of WSJ10 and WSJ40.

We should keep in mind that (1) a treebank PCFG is not state-of-the-art: its performance is mediocre compared to e.g. Bod (2003) or McClosky et al. (2006), and (2) that our treebank PCFG is binarized as in Klein and Manning (2005) to make results comparable. To be sure, the unbinarized version of the treebank PCFG obtains 89.0% average f-score on WSJ10 and 72.3% average f-score on WSJ40. Remember that the Penn Treebank annotations are often exceedingly flat, and many branches have arity larger than two. It would be interesting to see how UML-DOP performs if we also accept ternary (and wider) branches -- though the total number of possible trees that can be assigned to strings would then further explode.

UML-DOP's performance still remains behind that of *supervised* (binarized) DOP parsers, such as DOP1, which achieved 81.9% average f-score on the 10 WSJ40 splits, and ML-DOP, which performed slightly better with 82.1% average f-score. And if DOP1 and ML-DOP are not binarized, their average f-scores are respectively 90.1% and 90.5% on WSJ40. However, DOP1 and ML-DOP heavily depend on annotated data whereas UML-DOP only needs unannotated data. It would thus be interesting to see how close UML-DOP can get to ML-DOP's performance if we enlarge the amount of training data.

7 Conclusion: Is the end of supervised parsing in sight?

Now that we have outperformed a well-known supervised parser by an unsupervised one, we may raise the question as to whether the end of supervised NLP comes in sight. All supervised parsers are reaching an asymptote and further improvement does not seem to come from more hand-annotated data but by adding unsupervised or semi-unsupervised techniques (cf. McClosky et al. 2006). Thus if we modify our question as: does the *exclusively* supervised approach to parsing come to an end, we believe that the answer is certainly yes.

Yet we should neither rule out the possibility that entirely unsupervised methods will in fact surpass semi-supervised methods. The main problem is how to quantitatively compare these different parsers, as any evaluation on hand-annotated data (like the Penn treebank) will unreasonably favor semi-supervised parsers. There is thus is a quest for designing an annotation-independent evaluation scheme. Since parsers are becoming increasingly important in applications like syntax-based machine translation and structural language models for speech recognition, one way to go would be to compare these different parsing methods by isolating their contribution in improving a concrete NLP system, rather than by testing them against gold standard annotations which are inherently theory-dependent.

The initially disappointing results of inducing trees entirely from raw text was not so much due to the difficulty of the bootstrapping problem *per se*, but to (1) the poverty of the initial models and (2) the difficulty of finding theory-independent evaluation criteria. The time has come to fully reappraise unsupervised parsing models which should be trained on massive amounts of data, and be evaluated in a concrete application.

There is a final question as to how far the DOP approach to unsupervised parsing can be stretched. In principle we can assign all possible syntactic categories, semantic roles, argument

structures etc. to a set of given sentences and let the statistics decide which assignments are most useful in parsing new sentences. Whether such a massively maximalist approach is feasible can only be answered by empirical investigation in due time.

Acknowledgements

Thanks to Willem Zuidema, David Tugwell and especially to three anonymous reviewers whose unanymous suggestions on DOP and EM considerably improved the original paper. A substantial part of this research was carried out in the context of the NWO Exact project "Unsupervised Stochastic Grammar Induction from Unlabeled Data", project number 612.066.405.

References

Bod, R. 1998. *Beyond Grammar: An Experience-Based Theory of Language*, CSLI Publications, distributed by Cambridge University Press.

Bod, R. 2000. Combining semantic and syntactic structure for language modeling. *Proceedings ICSLP 2000*, Beijing.

Bod, R. 2003. An efficient implementation of a new DOP model. *Proceedings EACL 2003*, Budapest.

Bod, R. 2006. Unsupervised Parsing with U-DOP. *Proceedings CONLL 2006*, New York.

Bonnema, R., R. Bod and R. Scha, 1997. A DOP model for semantic interpretation, *Proceedings ACL/EACL 1997*, Madrid.

Chi, Z. and S. Geman 1998. Estimation of Probabilistic Context-Free Grammars. *Computational Linguistics* 24, 299-305.

Clark, A. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. *Proceedings CONLL 2001*.

Collins, M. and N. Duffy 2002. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. *Proceedings ACL 2002*, Philadelphia.

Dempster, A., N. Laird and D. Rubin, 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm, *Journal of the Royal Statistical Society* 39, 1-38.

Goodman, J. 2003. Efficient algorithms for the DOP model. In R. Bod, R. Scha and K. Sima'an (eds.). *Data-Oriented Parsing*, University of Chicago Press.

Huang, L. and D. Chiang 2005. Better *k*-best parsing. *Proceedings IWPT 2005*, Vancouver.

Johnson, M. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics* 28, 71-76.

Klein, D. 2005. *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University.

Klein, D. and C. Manning 2002. A general constituent-context model for improved grammar induction. *Proceedings ACL 2002*, Philadelphia.

Klein, D. and C. Manning 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. *Proceedings ACL 2004*, Barcelona.

Klein, D. and C. Manning 2005. Natural language grammar induction with a generative constituent-context model. *Pattern Recognition* 38, 1407-1419.

Magerman, D. 1993. *Expectation-Maximization for Data-Oriented Parsing*, IBM Technical Report, Yorktown Heights, NY.

McClosky, D., E. Charniak and M. Johnson 2006. Effective self-training for parsing. *Proceedings HLT-NAACL 2006*, New York.

Manning, C. and H. Schütze 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Prescher, D., R. Scha, K. Sima'an and A. Zollmann 2004. On the statistical consistency of DOP estimators. *Proceedings CLIN 2004*, Leiden.

Schütze, H. 1995. Distributional part-of-speech tagging. *Proceedings ACL 1995*, Dublin.

Shao, J. 1999. *Mathematical Statistics*. Springer Verlag, New York.

Sima'an, K. 1996. Computational complexity of probabilistic disambiguation by means of tree grammars. *Proceedings COLING 1996*, Copenhagen.

Skut, W., B. Krenn, T. Brants and H. Uszkoreit 1997. An annotation scheme for free word order languages. *Proceedings ANLP 1997.*

Xue, N., F. Chiou and M. Palmer 2002. Building a large-scale annotated Chinese corpus. *Proceedings COLING 2002*, Taipei.

van Zaanen, M. 2000. ABL: Alignment-Based Learning. *Proceedings COLING 2000*, Saarbrücken.

Zollmann, A. and K. Sima'an 2005. A consistent and efficient estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics*, in press.

Zuidema, W. 2006. What are the productive units of natural language grammar? A DOP approach to the automatic identification of constructions. *Proceedings CONLL 2006*, New York.

Is the End of Supervised Parsing in Sight?

Rens Bod
School of Computer Science
University of St Andrews, ILLC, University of Amsterdam
rb@cs.st-and.ac.uk

Abstract

How far can we get with unsupervised parsing if we make our training corpus several orders of magnitude larger than has hitherto be attempted? We present a new algorithm for unsupervised parsing using an all-subtrees model, termed U-DOP*, which parses directly with packed forests of all binary trees. We train both on Penn's WSJ data and on the (much larger) NANC corpus, showing that U-DOP* outperforms a treebank-PCFG on the standard WSJ test set. While U-DOP* performs worse than state-of-the-art supervised parsers on hand-annotated sentences, we show that the model outperforms supervised parsers when evaluated as a language model in syntax-based machine translation on Europarl. We argue that supervised parsers miss the fluidity between constituents and non-constituents and that in the field of syntax-based language modeling the end of supervised parsing has come in sight.

1 Introduction

A major challenge in natural language parsing is the unsupervised induction of syntactic structure. While most parsing methods are currently supervised or semi-supervised (McClosky et al. 2006; Henderson 2004; Steedman et al. 2003), they depend on hand-annotated data which are difficult to come by and which exist only for a few languages. Unsupervised parsing methods are becoming increasingly important since they operate with raw, unlabeled data of which unlimited quantities are available.

There has been a resurgence of interest in unsupervised parsing during the last few years. Where van Zaanen (2000) and Clark (2001) induced unlabeled phrase structure for small domains like the ATIS, obtaining around 40% unlabeled f-score, Klein and Manning (2002) report 71.1% f-score on Penn WSJ part-of-speech strings ≤ 10 words (WSJ10) using a constituent-context model called CCM. Klein and Manning (2004) further show that a hybrid approach which combines constituency and dependency models, yields 77.6% f-score on WSJ10.

While Klein and Manning's approach may be described as an "all-substrings" approach to unsupervised parsing, an even richer model consists of an "all-subtrees" approach to unsupervised parsing, called U-DOP (Bod 2006). U-DOP initially assigns all unlabeled binary trees to a training set, efficiently stored in a packed forest, and next trains subtrees thereof on a held-out corpus, either by taking their relative frequencies, or by iteratively training the subtree parameters using the EM algorithm (referred to as "UML-DOP"). The main advantage of an all-subtrees approach seems to be the direct inclusion of *discontiguous* context that is not captured by (linear) substrings. Discontiguous context is important not only for learning structural dependencies but also for learning a variety of non-contiguous constructions such as *nearest … to…* or *take … by surprise*. Bod (2006) reports 82.9% unlabeled f-score on the same WSJ10 as used by Klein and Manning (2002, 2004). Unfortunately, his experiments heavily depend on a priori sampling of subtrees, and the model becomes highly inefficient if larger corpora are used or longer sentences are included.

In this paper we will also test an alternative model for unsupervised all-subtrees

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 400–407,
Prague, Czech Republic, June 2007.

parsing, termed U-DOP*, which is based on the DOP* estimator by Zollmann and Sima'an (2005), and which computes the shortest derivations for sentences from a held-out corpus using all subtrees from all trees from an extraction corpus. While we do not achieve as high an f-score as the UML-DOP model in Bod (2006), we will show that U-DOP* can operate without subtree sampling, and that the model can be trained on corpora that are two orders of magnitude larger than in Bod (2006). We will extend our experiments to 4 million sentences from the NANC corpus (Graff 1995), showing that an f-score of 70.7% can be obtained on the standard Penn WSJ test set by means of unsupervised parsing. Moreover, U-DOP* can be directly put to use in bootstrapping structures for concrete applications such as syntax-based machine translation and speech recognition. We show that U-DOP* outperforms the supervised DOP model if tested on the German-English Europarl corpus in a syntax-based MT system.

In the following, we first explain the DOP* estimator and discuss how it can be extended to unsupervised parsing. In section 3, we discuss how a PCFG reduction for supervised DOP can be applied to packed parse forests. In section 4, we will go into an experimental evaluation of U-DOP* on annotated corpora, while in section 5 we will evaluate U-DOP* on unlabeled corpora in an MT application.

2 From DOP* to U-DOP*

DOP* is a modification of the DOP model in Bod (1998) that results in a statistically consistent estimator and in an efficient training procedure (Zollmann and Sima'an 2005). DOP* uses the all-subtrees idea from DOP: given a treebank, take all subtrees, regardless of size, to form a stochastic tree-substitution grammar (STSG). Since a parse tree of a sentence may be generated by several (leftmost) derivations, the probability of a tree is the sum of the probabilities of the derivations producing that tree. The probability of a derivation is the product of the subtree probabilities. The original DOP model in Bod (1998) takes the occurrence frequencies of the subtrees in the trees normalized by their root frequencies as subtree parameters. While efficient algorithms have been developed for this DOP model by converting it into a PCFG reduction (Goodman 2003), DOP's estimator was shown to be inconsistent by Johnson (2002). That is, even with unlimited training data, DOP's estimator is not guaranteed to converge to the correct distribution.

Zollmann and Sima'an (2005) developed a statistically consistent estimator for DOP which is based on the assumption that maximizing the joint probability of the parses in a treebank can be approximated by maximizing the joint probability of their shortest derivations (i.e. the derivations consisting of the fewest subtrees). This assumption is in consonance with the principle of simplicity, but there are also empirical reasons for the shortest derivation assumption: in Bod (2003) and Hearne and Way (2006), it is shown that DOP models that select the preferred parse of a test sentence using the shortest derivation criterion perform very well.

On the basis of this shortest-derivation assumption, Zollmann and Sima'an come up with a model that uses held-out estimation: the training corpus is randomly split into two parts proportional to a fixed ratio: an *extraction corpus* EC and a *held-out* corpus HC. Applied to DOP, held-out estimation would mean to extract fragments from the trees in EC and to assign their weights such that the likelihood of HC is maximized. If we combine their estimation method with Goodman's reduction of DOP, Zollman and Sima'an's procedure operates as follows:

(1) Divide a treebank into an EC and HC
(2) Convert the subtrees from EC into a PCFG reduction
(3) Compute the shortest derivations for the sentences in HC (by simply assigning each subtree equal weight and applying Viterbi 1-best)
(4) From those shortest derivations, extract the subtrees and their relative frequencies in HC to form an STSG

Zollmann and Sima'an show that the resulting estimator is consistent. But equally important is the fact that this new DOP* model does not suffer from a decrease in parse accuracy if larger subtrees are included, whereas the original DOP model needs to be redressed by a correction factor to maintain this property (Bod 2003). Moreover, DOP*'s estimation procedure is very efficient, while the EM training procedure for UML-DOP

proposed in Bod (2006) is particularly time consuming and can only operate by randomly sampling trees.

Given the advantages of DOP*, we will generalize this model in the current paper to *unsupervised* parsing. We will use the same all-subtrees methodology as in Bod (2006), but now by applying the efficient and consistent DOP*-based estimator. The resulting model, which we will call U-DOP*, roughly operates as follows:

(1) Divide a corpus into an EC and HC
(2) Assign all unlabeled binary trees to the sentences in EC, and store them in a shared parse forest
(3) Convert the subtrees from the parse forests into a compact PCFG reduction (see next section)
(4) Compute the shortest derivations for the sentences in HC (as in DOP*)
(5) From those shortest derivations, extract the subtrees and their relative frequencies in HC to form an STSG
(6) Use the STSG to compute the most probable parse trees for new test data by means of Viterbi *n*-best (see next section)

We will use this U-DOP* model to investigate our main research question: *how far can we get with unsupervised parsing if we make our training corpus several orders of magnitude larger than has hitherto be attempted?*

3 Converting shared parse forests into PCFG reductions

The main computational problem is how to deal with the immense number of subtrees in U-DOP*. There exists already an efficient *supervised* algorithm that parses a sentence by means of all subtrees from a treebank. This algorithm was extensively described in Goodman (2003) and converts a DOP-based STSG into a compact PCFG reduction that generates eight rules for each node in the treebank. The reduction is based on the following idea: every node in every treebank tree is assigned a unique number which is called its address. The notation $A@k$ denotes the node at address k where A is the nonterminal labeling that node. A new nonterminal is created for each node in the training data. This nonterminal is called A_k. Let a_j represent the number of subtrees headed by the node $A@j$, and let a represent the number of subtrees headed by nodes with nonterminal A, that is $a = \Sigma_j\ a_j$. Then there is a PCFG with the following property: for every subtree in the training corpus headed by A, the grammar will generate an isomorphic subderivation. For example, for a node $(A@j\ (B@k,\ C@l))$, the following eight PCFG rules in figure 1 are generated, where the number following a rule is its weight.

$A_j \rightarrow BC$	$(1/a_j)$	$A \rightarrow BC$	$(1/a)$
$A_j \rightarrow B_kC$	(b_k/a_j)	$A \rightarrow B_kC$	(b_k/a)
$A_j \rightarrow BC_l$	(c_l/a_j)	$A \rightarrow BC_l$	(c_l/a)
$A_j \rightarrow B_kC_l$	(b_kc_l/a_j)	$A \rightarrow B_kC_l$	(b_kc_l/a)

Figure 1. PCFG reduction of supervised DOP

By simple induction it can be shown that this construction produces PCFG derivations isomorphic to DOP derivations (Goodman 2003: 130-133). The PCFG reduction is linear in the number of nodes in the corpus.

While Goodman's reduction method was developed for supervised DOP where each training sentence is annotated with exactly one tree, the method can be generalized to a corpus where each sentence is annotated with all possible binary trees (labeled with the generalized category X), as long as we represent these trees by a shared parse forest. A shared parse forest can be obtained by adding pointers from each node in the chart (or tabular diagram) to the nodes that caused it to be placed in the chart. Such a forest can be represented in cubic space and time (see Billot and Lang 1989). Then, instead of assigning a unique address to each node in each tree, as done by the PCFG reduction for supervised DOP, we now assign a unique address to each node in each parse forest for each sentence. However, the same node may be part of more than one tree. A shared parse forest is an AND-OR graph where AND-nodes correspond to the usual parse tree nodes, while OR-nodes correspond to distinct subtrees occurring in the same context. The total number of nodes is cubic in sentence length n. This means that there are $O(n^3)$ many nodes that receive a unique address as described above, to which next our PCFG reduction is applied. This is a huge reduction compared to Bod (2006) where

the number of subtrees of all trees increases with the Catalan number, and only ad hoc sampling could make the method work.

Since U-DOP* computes the shortest derivations (in the training phase) by combining subtrees from unlabeled binary trees, the PCFG reduction in figure 1 can be represented as in figure 2, where *X* refers to the generalized category while *B* and *C* either refer to part-of-speech categories or are equivalent to *X*. The equal weights follow from the fact that the shortest derivation is equivalent to the most probable derivation if all subtrees are assigned equal probability (see Bod 2000; Goodman 2003).

$X_j \rightarrow BC$	1	$X \rightarrow BC$	0.5
$X_j \rightarrow B_kC$	1	$X \rightarrow B_kC$	0.5
$X_j \rightarrow BC_l$	1	$X \rightarrow BC_l$	0.5
$X_j \rightarrow B_kC_l$	1	$X \rightarrow B_kC_l$	0.5

Figure 2. PCFG reduction for U-DOP*

Once we have parsed HC with the shortest derivations by the PCFG reduction in figure 2, we extract the subtrees from HC to form an STSG. The number of subtrees in the shortest derivations is linear in the number of nodes (see Zollmann and Sima'an 2005, theorem 5.2). This means that U-DOP* results in an STSG which is much more succinct than previous DOP-based STSGs. Moreover, as in Bod (1998, 2000), we use an extension of Good-Turing to smooth the subtrees and to deal with 'unknown' subtrees.

Note that the direct conversion of parse forests into a PCFG reduction also allows us to efficiently implement the maximum likelihood extension of U-DOP known as UML-DOP (Bod 2006). This can be accomplished by training the PCFG reduction on the held-out corpus HC by means of the expectation-maximization algorithm, where the weights in figure 1 are taken as initial parameters. Both U-DOP*'s and UML-DOP's estimators are known to be statistically consistent. But while U-DOP*'s training phase merely consists of the computation of the shortest derivations and the extraction of subtrees, UML-DOP involves iterative training of the parameters.

Once we have extracted the STSG, we compute the most probable parse for new sentences by Viterbi *n*-best, summing up the probabilities of derivations resulting in the same tree (the exact computation of the most probable parse is NP hard – see Sima'an 1996). We have incorporated the technique by Huang and Chiang (2005) into our implementation which allows for efficient Viterbi *n*-best parsing.

4 Evaluation on hand-annotated corpora

To evaluate U-DOP* against UML-DOP and other unsupervised parsing models, we started out with three corpora that are also used in Klein and Manning (2002, 2004) and Bod (2006): Penn's WSJ10 which contains 7422 sentences ≤ 10 words after removing empty elements and punctuation, the German NEGRA10 corpus and the Chinese Treebank CTB10 both containing 2200+ sentences ≤ 10 words after removing punctuation. As with most other unsupervised parsing models, we train and test on p-o-s strings rather than on word strings. The extension to word strings is straightforward as there exist highly accurate unsupervised part-of-speech taggers (e.g. Schütze 1995) which can be directly combined with unsupervised parsers, but for the moment we will stick to p-o-s strings (we will come back to word strings in section 5). Each corpus was divided into 10 training/test set splits of 90%/10% (*n*-fold testing), and each training set was randomly divided into two equal parts, that serve as EC and HC and vice versa. We used the same evaluation metrics for unlabeled precision (UP) and unlabeled recall (UR) as in Klein and Manning (2002, 2004). The two metrics of UP and UR are combined by the unlabeled f-score F1 = 2*UP*UR/(UP+UR). All trees in the test set were binarized beforehand, in the same way as in Bod (2006).

For UML-DOP the decrease in cross-entropy became negligible after maximally 18 iterations. The training for U-DOP* consisted in the computation of the shortest derivations for the HC from which the subtrees and their relative frequencies were extracted. We used the technique in Bod (1998, 2000) to include 'unknown' subtrees. Table 1 shows the f-scores for U-DOP* and UML-DOP against the f-scores for U-DOP reported in Bod (2006), the CCM model in Klein and Manning (2002), the DMV dependency model in Klein and Manning (2004) and their combined model DMV+CCM.

Model	English (WSJ10)	German (NEGRA10)	Chinese (CTB10)
CCM	71.9	61.6	45.0
DMV	52.1	49.5	46.7
DMV+CCM	77.6	63.9	43.3
U-DOP	78.5	65.4	46.6
U-DOP*	77.9	63.8	42.8
UML-DOP	79.4	65.2	45.0

Table 1. F-scores of U-DOP* and UML-DOP compared to other models on the same data.

It should be kept in mind that an exact comparison can only be made between U-DOP* and UML-DOP in table 1, since these two models were tested on 90%/10% splits, while the other models were applied to the full WSJ10, NEGRA10 and CTB10 corpora. Table 1 shows that U-DOP* performs worse than UML-DOP in all cases, although the differences are small and was statistically significant only for WSJ10 using paired *t*-testing.

As explained above, the main advantage of U-DOP* over UML-DOP is that it works with a more succinct grammar extracted from the shortest derivations of HC. Table 2 shows the size of the grammar (number of rules or subtrees) of the two models for resp. Penn WSJ10, the entire Penn WSJ and the first 2 million sentences from the NANC (North American News Text) corpus which contains a total of approximately 24 million sentences from different news sources.

Model	Size of STSG for WSJ10	Size of STSG for Penn WSJ	Size of STSG for 2,000K NANC
U-DOP*	2.2×10^4	9.8×10^5	7.2×10^6
UML-DOP	1.5×10^6	8.1×10^7	5.8×10^9

Table 2. Grammar size of U-DOP* and UML-DOP for WSJ10 (7,7K sentences), WSJ (50K sentences) and the first 2,000K sentences from NANC.

Note that while U-DOP* is about 2 orders of magnitudes smaller than UML-DOP for the WSJ10, it is almost 3 orders of magnitudes smaller for the first 2 million sentences of the NANC corpus. Thus even if U-DOP* does not give the highest f-score in table 1, it is more apt to be trained on larger data sets. In fact, a well-known advantage of unsupervised methods over supervised methods is the availability of almost unlimited amounts of text. Table 2 indicates that U-DOP*'s grammar is still of manageable size even for text corpora that are (almost) two orders of magnitude larger than Penn's WSJ. The NANC corpus contains approximately 2 million WSJ sentences that do not overlap with Penn's WSJ and has been previously used by McClosky et al. (2006) in improving a supervised parser by self-training. In our experiments below we will start by mixing subsets from the NANC's WSJ data with Penn's WSJ data. Next, we will do the same with 2 million sentences from the LA Times in the NANC corpus, and finally we will mix all data together for inducing a U-DOP* model. From Penn's WSJ, we only use sections 2 to 21 for training (just as in supervised parsing) and section 23 (≤100 words) for testing, so as to compare our unsupervised results with some binarized supervised parsers.

The NANC data was first split into sentences by means of a simple discriminitive model. It was next p-o-s tagged with the the TnT tagger (Brants 2000) which was trained on the Penn Treebank such that the same tag set was used. Next, we added subsets of increasing size from the NANC p-o-s strings to the 40,000 Penn WSJ p-o-s strings. Each time the resulting corpus was split into two halfs and the shortest derivations were computed for one half by using the PCFG-reduction from the other half and vice versa. The resulting trees were used for extracting an STSG which in turn was used to parse section 23 of Penn's WSJ. Table 3 shows the results.

# sentences added	f-score by adding WSJ data	f-score by adding LA Times data
0 (baseline)	62.2	62.2
100k	64.7	63.0
250k	66.2	63.8
500k	67.9	64.1
1,000k	68.5	64.6
2,000k	69.0	64.9

Table 3. Results of U-DOP* on section 23 from Penn's WSJ by adding sentences from NANC's WSJ and NANC's LA Times

Table 3 indicates that there is a monotonous increase in f-score on the WSJ test set if NANC text is added to our training data in both cases, independent of whether the sentences come from the WSJ domain or the LA Times domain. Although the effect of adding LA Times data is weaker than adding WSJ data, it is noteworthy that the unsupervised induction of trees from the LA Times domain still improves the f-score even if the test data are from a different domain.

We also investigated the effect of adding the LA Times data to the total mix of Penn's WSJ and NANC's WSJ. Table 4 shows the results of this experiment, where the baseline of 0 sentences thus starts with the 2,040k sentences from the combined Penn-NANC WSJ data.

Sentences added from LA Times to Penn-NANC WSJ	f-score by adding LA Times data
0	69.0
100k	69.4
250k	69.9
500k	70.2
1,000k	70.4
2,000k	70.7

Table 4. Results of U-DOP* on section 23 from Penn's WSJ by mixing sentences from the combined Penn-NANC WSJ with additions from NANC's LA Times.

As seen in table 4, the f-score continues to increase even when adding LA Times data to the large combined set of Penn-NANC WSJ sentences. The highest f-score is obtained by adding 2,000k sentences, resulting in a total training set of 4,040k sentences. We believe that our result is quite promising for the future of unsupervised parsing.

In putting our best f-score in table 4 into perspective, it should be kept in mind that the gold standard trees from Penn-WSJ section 23 were binarized. It is well known that such a binarization has a negative effect on the f-score. Bod (2006) reports that an unbinarized treebank grammar achieves an average 72.3% f-score on WSJ sentences ≤ 40 words, while the binarized version achieves only 64.6% f-score. To compare U-DOP*'s results against some supervised parsers, we additionally evaluated a PCFG treebank grammar and the supervised DOP* parser using the same test set. For these supervised parsers, we employed the standard training set, i.e. Penn's WSJ sections 2-21, but only by taking the p-o-s strings as we did for our unsupervised U-DOP* model. Table 5 shows the results of this comparison.

Parser	f-score
U-DOP*	70.7
Binarized treebank PCFG	63.5
Binarized DOP*	80.3

Table 5. Comparison between the (best version of) U-DOP*, the supervised treebank PCFG and the supervised DOP* for section 23 of Penn's WSJ

As seen in table 5, U-DOP* outperforms the binarized treebank PCFG on the WSJ test set. While a similar result was obtained in Bod (2006), the absolute difference between unsupervised parsing and the treebank grammar was extremely small in Bod (2006): 1.8%, while the difference in table 5 is 7.2%, corresponding to 19.7% error reduction. Our f-score remains behind the supervised version of DOP* but the gap gets narrower as more training data is being added to U-DOP*.

5 Evaluation on unlabeled corpora in a practical application

Our experiments so far have shown that despite the addition of large amounts of unlabeled training data, U-DOP* is still outperformed by the supervised DOP* model when tested on hand-annotated corpora like the Penn Treebank. Yet it is well known that any evaluation on hand-annotated corpora unreasonably favors supervised parsers. There is thus a quest for designing an evaluation scheme that is independent of annotations. One way to go would be to compare supervised and unsupervised parsers as a syntax-based language model in a practical application such as machine translation (MT) or speech recognition.

In Bod (2007), we compared U-DOP* and DOP* in a syntax-based MT system known as Data-Oriented Translation or *DOT* (Poutsma 2000; Groves et al. 2004). The DOT model starts with a bilingual treebank where each tree pair constitutes an example translation and where translationally equivalent constituents are linked. Similar to DOP,

the DOT model uses all linked subtree pairs from the bilingual treebank to form an STSG of linked subtrees, which are used to compute the most probable translation of a target sentence given a source sentence (see Hearne and Way 2006).

What we did in Bod (2007) is to let both DOP* and U-DOP* compute the best trees directly for the *word strings* in the German-English Europarl corpus (Koehn 2005), which contains about 750,000 sentence pairs. Differently from U-DOP*, DOP* needed to be trained on annotated data, for which we used respectively the Negra and the Penn treebank. Of course, it is well-known that a supervised parser's f-score decreases if it is transferred to another domain: for example, the (non-binarized) WSJ-trained DOP model in Bod (2003) decreases from around 91% to 85.5% f-score if tested on the Brown corpus. Yet, this score is still considerably higher than the accuracy obtained by the unsupervised U-DOP model, which achieves 67.6% unlabeled f-score on Brown sentences. Our main question of interest is in how far this difference in accuracy on hand-annotated corpora carries over when tested in the context of a concrete application like MT. This is not a trivial question, since U-DOP* learns 'constituents' for word sequences such as *Ich möchte* ("I would like to") and *There are* (Bod 2007), which are usually hand-annotated as non-constituents. While U-DOP* is punished for this 'incorrect' prediction if evaluated on the Penn Treebank, it may be rewarded for this prediction if evaluated in the context of machine translation using the Bleu score (Papineni et al. 2002). Thus similar to Chiang (2005), U-DOP can discover non-syntactic phrases, or simply "phrases", which are typically neglected by linguistically syntax-based MT systems. At the same time, U-DOP* can also learn discontiguous constituents that are neglected by phrase-based MT systems (Koehn et al. 2003).

In our experiments, we used both U-DOP* and DOP* to predict the best trees for the German-English Europarl corpus. Next, we assigned links between each two nodes in the respective trees for each sentence pair. For a 2,000 sentence test set from a different part of the Europarl corpus we computed the most probable target sentence (using Viterbi *n* best). The Bleu score was used to measure translation accuracy, calculated by the NIST script with its default settings. As a baseline we compared our results with the publicly available phrase-based system Pharaoh (Koehn et al. 2003), using the default feature set. Table 6 shows for each system the Bleu score together with a description of the productive units. 'U-DOT' refers to 'Unsupervised DOT' based on U-DOP*, while DOT is based on DOP*.

System	Productive Units	Bleu-score
U-DOT / U-DOP*	Constituents and Phrases	0.280
DOT / DOP*	Constituents only	0.221
Pharaoh	Phrases only	0.251

Table 6. Comparing U-DOP* and DOP* in syntax-based MT on the German-English Europarl corpus against the Pharaoh system.

The table shows that the unsupervised U-DOT model outperforms the supervised DOT model with 0.059. Using Zhang's significance tester (Zhang et al. 2004), it turns out that this difference is statistically significant ($p < 0.001$). Also the difference between U-DOT and the baseline Pharaoh is statistically significant ($p < 0.008$). Thus even if supervised parsers like DOP* outperform unsupervised parsers like U-DOP* on hand-parsed data with >10%, the same supervised parser is outperformed by the unsupervised parser if tested in an MT application. Evidently, U-DOP's capacity to capture both constituents and phrases pays off in a concrete application and shows the shortcomings of models that only allow for either constituents (such as linguistically syntax-based MT) or phrases (such as phrase-based MT). In Bod (2007) we also show that U-DOT obtains virtually the same Bleu score as Pharaoh after eliminating subtrees with discontiguous yields.

6 Conclusion: future of supervised parsing

In this paper we have shown that the accuracy of unsupervised parsing under U-DOP* continues to grow when enlarging the training set with additional data. However, except for the simple treebank PCFG, U-DOP* scores worse than supervised parsers if evaluated on hand-annotated data. At the same time U-DOP* significantly outperforms the supervised DOP* if evaluated in a practical application like MT. We argued that this can be explained by the fact that U-DOP learns

both constituents and (non-syntactic) phrases while supervised parsers learn constituents only.

What should we learn from these results? We believe that parsing, when separated from a task-based application, is mainly an academic exercise. If we only want to mimick a treebank or implement a linguistically motivated grammar, then supervised, grammar-based parsers are preferred to unsupervised parsers. But if we want to improve a practical application with a syntax-based language model, then an unsupervised parser like U-DOP* might be superior.

The problem with most supervised (and semi-supervised) parsers is their rigid notion of constituent which excludes 'constituents' like the German *Ich möchte* or the French *Il y a*. Instead, it has become increasingly clear that the notion of constituent is a fluid which may sometimes be in agreement with traditional syntax, but which may just as well be in opposition to it. Any sequence of words can be a unit of combination, including non-contiguous word sequences like *closest* X *to* Y. A parser which does not allow for this fluidity may be of limited use as a language model. Since supervised parsers seem to stick to categorical notions of constituent, we believe that in the field of syntax-based language models the end of supervised parsing has come in sight.

Acknowledgements

Thanks to Willem Zuidema and three anonymous reviewers for useful comments and suggestions on the future of supervised parsing.

References

Billot, S. and B. Lang, 1989. The Structure of Shared Forests in Ambiguous Parsing. In *ACL 1989*.

Bod, R. 1998. *Beyond Grammar: An Experience-Based Theory of Language*, CSLI Publications.

Bod, R. Parsing with the Shortest Derivation. In *COLING 2000*, Saarbruecken.

Bod, R. 2003. An efficient implementation of a new DOP model. In *EACL 2003*, Budapest.

Bod, R. 2006. An All-Subtrees Approach to Unsupervised Parsing. In *ACL-COLING 2006*, Sydney.

Bod, R. 2007. Unsupervised Syntax-Based Machine Translation. Submitted for publication.

Brants, T. 2000. TnT - A Statistical Part-of-Speech Tagger. In *ANLP 2000*.

Chiang, D. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *ACL 2005*, Ann Arbor.

Clark, A. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. In *CONLL 2001*.

Goodman, J. 2003. Efficient algorithms for the DOP model. In R. Bod, R. Scha and K. Sima'an (eds.). *Data-Oriented Parsing*, CSLI Publications.

Graff, D. 1995. *North American News Text Corpus*. Linguistic Data Consortium. LDC95T21.

Groves, D., M. Hearne and A. Way, 2004. Robust Sub-Sentential Alignment of Phrase-Structure Trees. In *COLING 2004*, Geneva.

Hearne, M and A. Way, 2006. Disambiguation Strategies for Data-Oriented Translation. *Proceedings of the 11th Conference of the European Association for Machine Translation*, Oslo.

Henderson, J. 2004. Discriminative training of a neural network statistical parser. In *ACL 2004,* Barcelona.

Huang, L. and D. Chiang 2005. Better *k*-best parsing. In *IWPT 2005*, Vancouver.

Johnson, M. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics* 28, 71-76.

Klein, D. and C. Manning 2002. A general constituent-context model for improved grammar induction. In *ACL 2002*, Philadelphia.

Klein, D. and C. Manning 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. *ACL 2004*, Barcelona.

Koehn, P., Och, F. J., and Marcu, D. 2003. Statistical phrase based translation. In *HLT-NAACL 2003*.

Koehn, P. 2005. Europarl: a parallel corpus for statistical machine translation. In *MT Summit 2005*.

McClosky, D., E. Charniak and M. Johnson 2006. Effective self-training for parsing. In *HLT-NAACL 2006*, New York.

Poutsma, A. 2000. Data-Oriented Translation. In *COLING 2000*, Saarbruecken.

Schütze, H. 1995. Distributional part-of-speech tagging. In *ACL 1995*, Dublin.

Sima'an, K. 1996. Computational complexity of probabilistic disambiguation by means of tree grammars. In *COLING 1996*, Copenhagen.

Steedman, M. M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. In *EACL 2003*, Budapest.

van Zaanen, M. 2000. ABL: Alignment-Based Learning. In *COLING 2000*, Saarbrücken.

Zhang, Y., S. Vogel and A. Waibel, 2004. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC).*

Zollmann, A. and K. Sima'an 2005. A consistent and efficient estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics*, Vol. 10 (2005) Number 2/3, 367-388.

From Exemplar to Grammar: Integrating Analogy and Probability in Language Learning

Rens Bod
Institute for Logic, Language and Computation
University of Amsterdam
rens@science.uva.nl

Prepublication Draft

Abstract

We present a new model of language learning which is based on the following idea: if a language learner does not know which phrase-structure trees should be assigned to initial sentences, s/he allows (implicitly) for all possible trees and lets linguistic experience decide which is the 'best' tree for each sentence. The best tree is obtained by maximizing 'structural analogy' between a sentence and previous sentences, which is formalized by the most probable shortest combination of subtrees from all trees of previous sentences. Corpus-based experiments with this model on the Penn Treebank and the Childes database indicate that it can learn both exemplar-based and rule-based aspects of language, ranging from phrasal verbs to auxiliary fronting. By having learned the syntactic structures of sentences, we have also learned the grammar implicit in these structures, which can in turn be used to produce new sentences. We show that our model mimicks children's language development from item-based constructions to abstract constructions, and that the model can simulate some of the errors made by children in producing complex questions.

1 Introduction

It used to be a cliché that humans produce and understand new utterances by constructing analogies with utterances they experienced previously.[1] A formal articulation of this idea was, however, lacking for a long time. Although the notion of analogy has been successfully worked out for phonology (e.g. MacWhinney 1978) and morphology (e.g. Skousen 1989), linguistic theory seems to have given up on the problem of developing a formal notion of *syntactic* analogy. Common wisdom has had it that analogy is intrinsically flawed for syntax where unlimited generative capacity is needed.

In Bod (1998) we argued that this common wisdom is wrong. We developed a model of syntactic analysis which derives new sentences by combining fragments from a corpus of previously derived sentences. This model, known as Data-Oriented Parsing (DOP) (Scha 1990), was general enough to be instantiated for various linguistic representations, such as lexical-functional grammar (Bod and Kaplan 1998), head-driven phrase-structure grammar (Neumann and Flickinger 2002) and tree-adjoining grammar (Hoogweg 2003). The original DOP model (Bod 1998) operates on simple phrase-structure trees and maximizes a notion of "structural analogy" between a sentence and a corpus of previous sentence-structures. That is, it produces a new sentence-structure out of largest and most frequent overlaps with structures of previously experienced sentences. The model could used both for sentence analysis and sentence generation. While the DOP approach was successful in some respects, for instance in modeling acceptability judgments (Bod 2001), ambiguity resolution (Scha et al. 1999) and construction learning (Borensztajn et al. 2008), it had an important shortcoming as well: The approach did not account for the acquisition of *initial* structures. The DOP approach assumes that the structures of previous linguistic experiences are given and stored in a corpus. As such, DOP can at best account for adult language, and has nothing to say about how these structures are acquired. While we conjectured in Bod (2006a) that the approach can be extended to language learning, we left a gap between the intuitive idea and its concrete instantiation.

In the current paper we want to start to close that gap. We propose a generalization of DOP, termed *U-DOP* ("Unsupervised DOP"), which starts with the notion of tree structure. Our cognitive claim is that if a language learner does not know which tree structures should be assigned to initially perceived sentences, s/he allows (implicitly) for all possible tree structures and lets linguistic experience decide which structures are most useful for parsing new input. Similar to DOP, U-DOP analyzes a new sentence out of largest and most frequent subtrees from trees of previous sentences. The fundamental difference with the supervised DOP approach is that U-DOP takes into account subtrees from *all* possible (binary) trees of previous sentences rather than from a set of manually annotated trees.

Although we do not claim that the U-DOP model in this paper provides any near-to-complete theory of language acquisition, we will show that it can learn various linguistic phenomena, ranging from phrasal verbs to auxiliary fronting. Once we have learned the syntactic structures of sentences, we have also learned the grammar implicit in these structures, which can be used to produce new sentences. We will test this implicit grammar against children's language production from the Childes database, which indicates that children learn discontiguous dependencies at a very early age. We will show that complex syntactic phenomena, such as auxiliary fronting, can be learned by U-DOP without having seen them in the linguistic input and

[1] Chomsky (1966) argues that he found this view in Bloomfield, Hockett, Paul, Saussure, Jespersen, and "many others". For an historical overview, see Esper (1973).

without assuming that they are hard-wired in the mind. Instead, we will demonstrate that phenomena such as auxiliary fronting can be learned from simpler sentences by means of structural analogy. We argue that our results may shed new light on the well-known Poverty of the Stimulus argument according to which linguistic evidence is hopelessly underdetermined such that innate prior knowledge is needed (Chomsky 1965, 1971).

In the following section, we will first give a review of Data-Oriented Parsing. In Section 3, we will show how DOP can be generalized to language learning, resulting in the U-DOP model. In Section 4, we show how the approach can accurately learn structures for adult language, and in Section 5, we will extend our experiments to child language from the Childes database showing that the model can simulate the incremental learning of separable particle verbs. We will generalize our approach to language generation in Section 6 and perform some experiments with producing complex yes/no questions with auxiliary fronting. We end with a conclusion in Section 7.

2 Review of DOP: integrating rules and exemplars

One of the main motivations behind the DOP framework was to integrate rule-based and exemplar-based approaches to language processing (Scha 1990; Bod 1992, 1998; Kaplan 1996; Zollmann and Sima'an 2005; Zuidema 2006). While rules or generalizations are typically the building blocks in grammar-based theories of language (Chomsky 1965; Pinker 1999), exemplars or "stored linguistic tokens" are taken to be the primitives in usage-based theories (cf. Barlow and Kemmer 2000; Bybee 2006). However, several researchers have emphasized that both rules and exemplars play a role in language use and acquisition (Langacker 1987; Goldberg 2006; Abbott-Smith and Tomasello 2006). The DOP model is consonant with this view but takes it one step further: It proposes that rules and exemplars are part of the same distribution, and that both can be represented by *subtrees* from a corpus of tree structures of previously encountered sentences (Bod 2006a). DOP uses these subtrees as the productive units by which new sentences are produced and understood. The smallest subtrees in DOP correspond to the traditional notion of phrase-structure rule, while the largest subtrees correspond to full phrase-structure trees. But DOP also takes into account the middle ground between these two extremes which consists of all intermediate subtrees that are larger than phrase-structure rules and smaller than full sentence-structures.

To give a very simple example, assume that the phrase-structure tree for *Mary saw John* in Figure 1 constitutes our entire corpus. Then the set of all subtrees from this corpus is given in Figure 2.

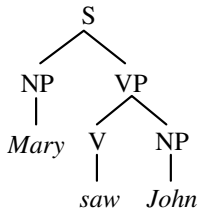


Figure 1. Phrase structure tree for *Mary saw John*

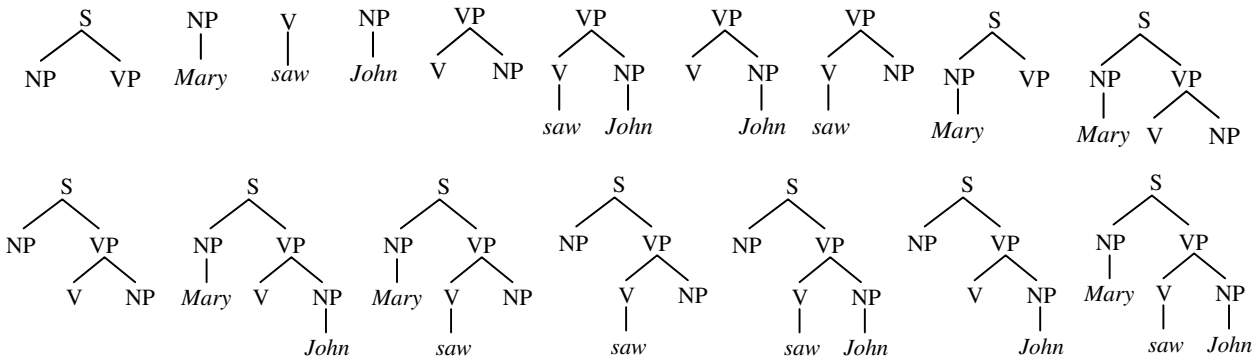


Figure 2. Subtrees from the tree in Figure 1

Thus the top-leftmost subtree in Figure 2 is equivalent to the traditional context-free rewrite rule *S* → *NP VP*, while the bottom-rightmost subtree corresponds to a phrase-structure tree for the entire sentence. But there is also a set of intermediate subtrees between these two endpoints that represent all other possible exemplars, such as *Mary V John*, *NP saw NP*, *Mary V NP*, etcera. The key idea of DOP which has been extensively argued for in Bod (1998) is the following: *Since we do not know beforehand which subtrees are important, we should not restrict them but take them all and let the statistics decide*. The DOP approach is thus congenial to the usage-based view of construction grammar where patterns are stored even if they are fully compositional (Croft 2001).

DOP generates new sentences by combining subtrees from a corpus of previously analyzed sentences. To illustrate this in some detail, consider a corpus of two sentences with their syntactic analyses given in Figure 3.

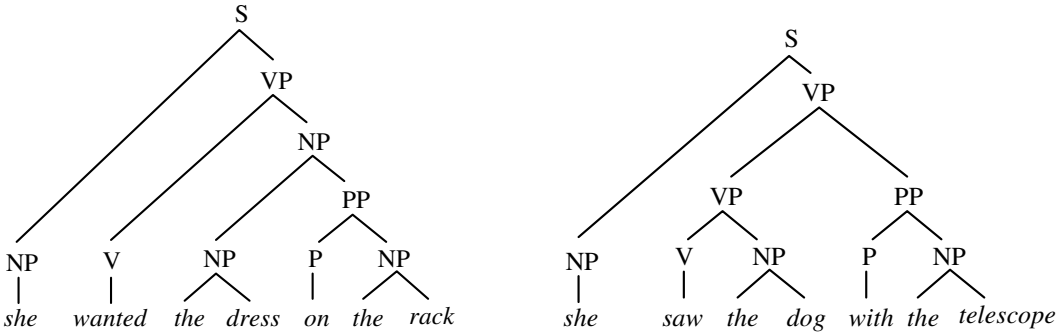


Figure 3. An extremely small corpus of two phrase-structure trees

On the basis of this corpus, the (new) sentence *She saw the dress with the telescope* can for example be derived by combining two subtrees from the corpus, as shown in Figure 4. The combination operation between subtrees is referred to as *label substitution*. This operation, indicated as ∘, identifies the leftmost nonterminal leaf node of the first subtree with the root node of the second subtree, i.e., the second subtree is substituted on the lefmost nonterminal leaf node of the first subtree provided that their categories match.

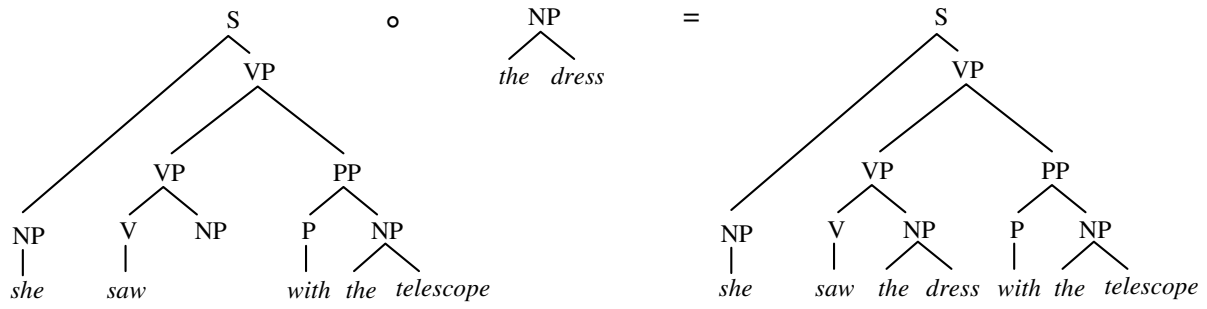


Figure 4. Analyzing a new sentence by combining subtrees from Figure 3

Notice that in Figure 4, the sentence *She saw the dress with the telescope* is interpreted analogously to the corpus sentence *She saw the dog with the telescope*: both sentences receive the same phrase structure where the prepositional phrase *with the telescope* is attached to the VP *saw the dress*.

We can also derive an alternative phrase structure for the test sentence, namely by combining three (rather than two) subtrees from Figure 3, as shown in Figure 5. We will write $(t \circ u) \circ v$ as $t \circ u \circ v$ with the convention that $\circ$ is *left*-associative.

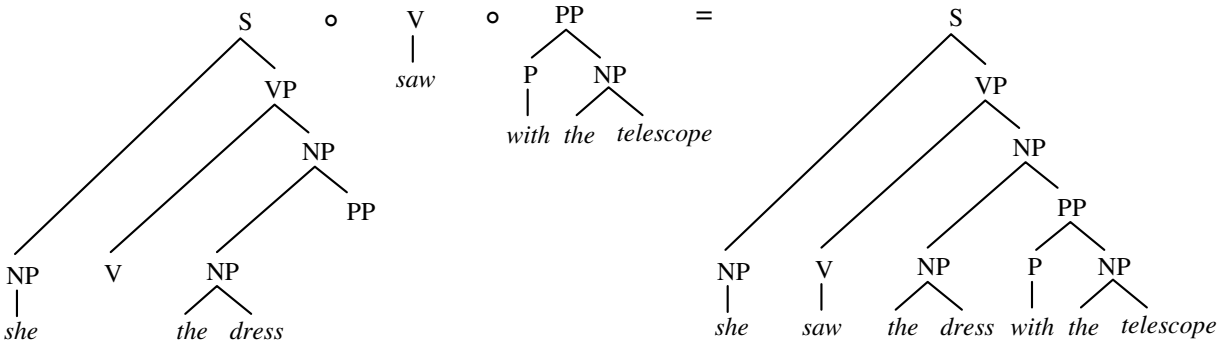


Figure 5. A different derivation for *She saw the dress with the telescope*

In Figure 5, the sentence *She saw the dress with the telescope* is analyzed in a different way where the PP *with the telescope* is attached to the NP *the dress*, corresponding to a different meaning than the tree in Figure 4. Thus the sentence is ambiguous in that it can be derived in (at least) two different ways which is analogous either to the first tree or to the second tree in Figure 3.

Note that an unlimited number of sentences can be generated by combining subtrees from the corpus in Figure 3, such as *She saw the dress on the rack with the telescope* and *She saw the dress with the dog on the rack with the telescope*, etc. Thus we obtain unlimited productivity by finite means. Note also that most sentences generated by this DOP model are highly ambiguous: many different analyses can be assigned to each sentence due to a combinatorial explosion of different prepositional-phrase attachments. Yet, most of the analyses are not plausible: They do not correspond to the interpretations humans perceive. There is thus a question how to rank different candidate-analyses of a sentence (or in case of generation, how to rank different candidate-sentences for a meaning to be conveyed). Initial DOP models proposed an exclusively frequency-based metric where the most probable tree or sentence was computed from the frequencies of the subtrees in the corpus (Bod 1998).

While it is well known that the frequency of a structure is an important factor in language

comprehension and production (see Jurafsky 2003), it is not the only factor. Discourse context, semantics and recency also play an important role. DOP can straightforwardly take into account semantic and discourse information if we have e.g. semantically annotated corpora from which we take the subtrees (Bonnema et al 1997). The notion of recency can furthermore be incorporated by a frequency-adjustment function which adjusts subtrees from recently perceived trees upwards while less recently perceived subtrees are adjusted downwards, possibly down to zero (Bod 1998, 1999).

There is, however, an important other factor which does not correspond to the notion of frequency: this is the *simplicity* of a structure (cf. Chater 1999). In Bod (2000 2002), we formalized the simplest structure by the *shortest derivation* of a sentence, i.e. consisting of the fewest subtrees from the corpus. Note that the shortest derivation will include the largest possible subtrees from the corpus, thereby *maximizing the structural commonality between a sentence and previous sentence-structures*. Only in case the shortest derivation is not unique, the frequencies of the subtrees are used to break ties. That is, DOP selects the tree with most frequent subtrees from the shortest derivations. This so-called 'best tree' of a sentence under DOP is defined as the Most Probable tree generated by the Shortest Derivation ("MPSD") of the sentence.

Rather than computing the most probable tree for a sentence *per se*, this model thus computes the most probable tree from among the distribution of trees that share maximal overlaps with previous sentence-structures. The MPSD maximizes what we call the *structural analogy* between a sentence and previous sentence-structures.[2] The shortest derivation may be seen as a formalization of the principle of 'least effort' or 'parsimony', while the notion of probability of a tree may be seen as a general memory-based frequency bias (cf. Conway and Christiansen 2006).

We can illustrate DOP's notion of structural analogy with the linguistic example given in the figures above. DOP predicts that the tree structure in Figure 4 is preferred because it can be generated by just two subtrees from the corpus. Any other tree structure, such as in Figure 5, would need at least three subtrees from the training set in Figure 3. Note that the tree generated by the shortest derivation indeed has a larger overlap with a corpus tree than the tree generated by the longer derivation.

Had we restricted the subtrees to smaller sizes -- for example to depth-1 subtrees, which makes DOP equivalent to a simple (probabilistic) context-free grammar -- the shortest derivation would not be able to distinguish between the two trees in Figures 3 and 5 as they would both be generated by 9 rewrite rules. The same is true if we used subtrees of maximal depth 2 or 3. As shown by Carroll and Weir (2000) only if we do not restrict the subtree depth, can we take into account arbitrarily far-ranging dependencies – both structurally and sequentially -- and model new sentences as closely as possible on previous sentence-analyses.

When the shortest derivation is not unique, DOP selects the tree with most frequent subtrees from the shortest derivations, i.e. the MPSD. Of course, even the MPSD may not be unique, in which case there is more than one best tree for the particular sentence; but such a situation does never occur in practice. In the following, we will define how the frequencies of the subtree that make up a parse tree can be compositionally combined to compute the MPSD. It is convenient to first give definitions for a parse tree under DOP and the shortest derivation.

[2] We prefer the term "analogy" to other terms like "similarity" since it reflects DOP's property to analyze a new sentence analogously to previous sentences, that is, DOP searches for relations between parts of a sentence(-structure) and corpus sentence(s) and maps the structure of previous sentences to new ones. This is consonant with the use of analogy in Gentner and Markman (1997).

Definition of a tree of a sentence generated by DOP
Given a corpus C of trees T_1, T_2,..,T_n, and a leftmost label substitution operation $\circ$, then a tree of a word string W with respect to C is a tree T such that (1) there are subtrees t_1, t_2,..,t_k in T_1, T_2,..,T_n for which $t_1 \circ t_2 \circ \ ... \circ t_k = T$, and (2) the yield of T is equal to W.

The tree generated by the shortest derivation, T_{sd} according to DOP is defined as follows:

Definition of the shortest derivation of a sentence
Let $L(d)$ be the length of derivation d in terms of its number of subtrees, that is, if $d = t_1 \circ ... \circ t_k$ then $L(d) = k$. Let d_T be a derivation which results in tree T. Then T_{sd} is the tree which is produced by a derivation of minimal length:

$$T_{sd} = \underset{T}{\text{argmin}}\, L(d_T)$$

If T_{sd} is not unique, DOP selects from among the trees produced by the shortest derivations the tree with highest probability. The probability of a tree is defined in terms of the probabilities of the derivations that generate it, which are in turn defined in terms of the probabilities of the subtrees these derivations consist of, as defined below.

Definition of the probability of a subtree
The probability of a subtree t, $P(t)$, is the number of occurrences of t in any tree in the corpus, written as $| t |$, divided by the total number of occurrences of subtrees in the corpus that have the same root label as t.[3] Let $r(t)$ return the root label of t. Then we may write:

$$P(t) = \frac{| t |}{\sum_{t':\ r(t')=r(t)} | t' |}$$

Definition of the probability of a derivation
The probability of a derivation $t_1 \circ ... \circ t_k$ is defined as the product of the probabilities of its subtrees t_i:

$$P(t_1 \circ ... \circ t_k) = \prod_i P(t_i)$$

Definition of the probability of a tree
Since DOP's subtrees can be of arbitrary size, it is typically the case that there are several derivations that generate the same parse tree. The probability of a tree T is defined as the sum of the probabilities of its distinct derivations. Let t_{id} be the i-th subtree in the derivation d that produces tree T, then the probability of T is given by

[3] The subtree probabilities are smoothed by simple Good-Turing estimation (see Bod 1998: 85).

$$P(T) = \sum_d \prod_i P(t_{id})$$

Definition of the best tree of a sentence
The best tree is the most probable tree from among the trees generated by the shortest derivation of a given sentence, also called the *MPSD*. The best tree, T_{best} maximizes the probability of T_{sd} given word string W:

$$T_{best} = \text{argmax}_{T_{sd}} P(T_{sd} \mid W)$$

We will give a concrete illustration of how the best tree can be computed in the following section when we generalize DOP to language acquisition. Although we have only dealt with the probabilities of derivations and trees, the model can also provide probabilities for each sentence generated by DOP, being the sum of the probabilities of all derivations generating that sentence. While DOP has mainly been applied to parsing, it was extended in Bonnema et al. (1997) to semantic interpretation and generation: given a meaning to be conveyed (e.g. a logical form), DOP's MPSD computes the best sentence for that meaning. We will come back to sentence generation in Section 6. The Appendix gives a summary of efficient algorithms for DOP.

Formally, the DOP model explained above is equivalent to a probabilistic tree-substitution grammar (PTSG). The grammatical backbone of a PTSG is a generalization over the well-known context-free grammars (CFG) and a subclass of Tree Adjoining Grammars (Joshi 2004). The original DOP model in Bod (1992), which only computed the most probable tree of each sentence (DOP1), had an inconsistent estimator: Johnson (2002) showed that the most probable trees do not converge to the correct trees when the corpus grows to infinity. However, Zollmann and Sima'an (2005) showed that a DOP model based on the shortest derivation is statistically consistent. Consistency is not to be confused with "tightness", i.e. the property that the total probability mass of the trees generated by a probabilistic grammar is equal to one (Chi and Geman 1998). Since DOP's PTSGs are weakly stochastically equivalent to so-called Treebank-PCFGs (Bod 1998), the probabilities of all trees for all sentences sum up to one (see Chi and Geman 1998).

3 U-DOP: generalizing DOP to language learning

In the current paper we generalize DOP to language learning by using the same principle as before: language users maximize the structural analogy between a new sentence and previous sentences by computing the most probable shortest derivation. However, in language learning we cannot assume that the phrase-structure trees of sentences are already given. We therefore propose the following straightforward generalization of DOP which we refer to as "Unsupervised DOP" or *U-DOP*: *if a language learner does not know which phrase-structure tree should be assigned to a sentence, s/he initially allows for all possible trees and let linguistic experience decide which is the 'best' tree by maximizing structural analogy*. As a first approximation we will limit the set of all possible trees to unlabeled binary trees. However, we can easily relax the binary restriction, and we will briefly come back to learning category labels at the end of this paper. Conceptually, we can distinguish three learning phases under U-DOP (though we will see that U-DOP operates rather differently from a computation point of view):

(i) Assign all possible (unlabeled binary) trees to a set of given sentences

(ii) Divide the binary trees into all subtrees
(iii) Compute the best tree (MPSD) for each sentence

The only prior knowledge assumed by U-DOP is the notion of tree and the concept of structural analogy (MPSD). U-DOP thus inherits the agnostic approach of DOP: we do not constrain the units of learning beforehand, but take all possible fragments and let a statistical notion of analogy decide.

In the following we will illustrate U-DOP with a simple example, by describing each of the three learning phases above separately.

(i) Assign all unlabeled binary trees to a set of sentences

Suppose that a hypothetical language learner hears the two sentences *watch the dog* and *the dog barks*. How could the learner figure out the appropriate tree structures for these sentences? U-DOP conjectures that a learner does so by allowing (initially) any fragment of the heard sentences to form a productive unit and to try to reconstruct these sentences out of most probable shortest combinations.

The set of all unlabeled binary trees for the sentences *watch the dog* and *the dog barks* is given in Figure 6, which for convenience we shall again refer to as the "corpus". Each node in each tree in the corpus is assigned the same category label *X*, since we do not (yet) know what label each phrase will receive. To keep our example simple, we do not assign category labels *X* to the words, but this can be done as well (and will be done later).

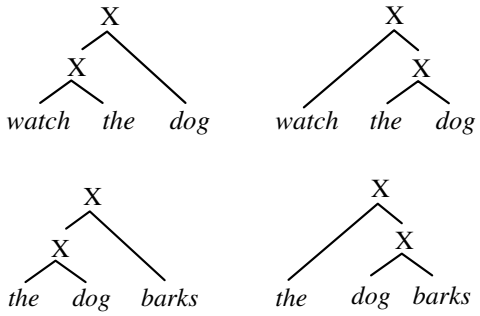


Figure 6. The unlabeled binary tree set for *watch the dog* and *the dog barks*

Although the number of possible binary trees for a sentence grows exponentially with sentence length, these binary trees can be efficiently represented in quadratic space by means of a "chart" or "tabular diagram", which is a standard technique in computational linguistics (see e.g. Kay 1980; Manning and Schütze 1999; Huang and Chiang 2005). By adding pointers between the nodes we obtain a structure known as a "shared parse forest" (Billot and Lang 1989). However, for explaining the conceptual working of U-DOP we will mostly exhaustively enumerate all trees, keeping in mind that the trees are usually stored by a compact parse forest.

(ii) Divide the binary trees into all subtrees

Figure 7 lists the subtrees that can be extracted from the trees in Figure 6. The first subtree in each row represents the whole sentence as a chunk, while the second and the third are "proper" subtrees.

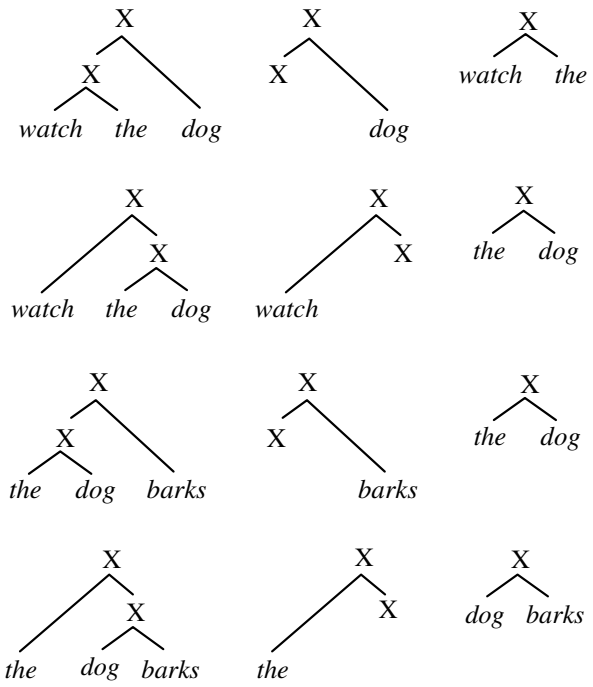


Figure 7. The subtree set for the binary trees in Figure 6.

Note that while most subtrees occur once, the subtree $[\textit{the dog}]_X$ occurs twice. The number of subtrees in a binary tree grows exponentially with sentence length, but there exists an efficient parsing algorithm that parses a sentence by means of all subtrees from a set of given trees. This algorithm converts a set of subtrees into a compact reduction which is linear in the number of tree nodes (Goodman 2003). We will come back to this reduction method below under (iii).

(iii) Compute the MPSD for each sentence

From the subtrees in Figure 7, U-DOP can compute the 'best trees' (MPSD) for the corpus sentences as well as for new sentences. Consider the corpus sentence *the dog barks*. On the basis of the subtrees in Figure 7, two phrase-structure trees can be generated by U-DOP for this sentence, shown in Figure 8. Both tree structures can be produced by two different derivations, either by trivially selecting the largest possible subtrees from Figure 7 that span the whole sentence or by combining two smaller subtrees.

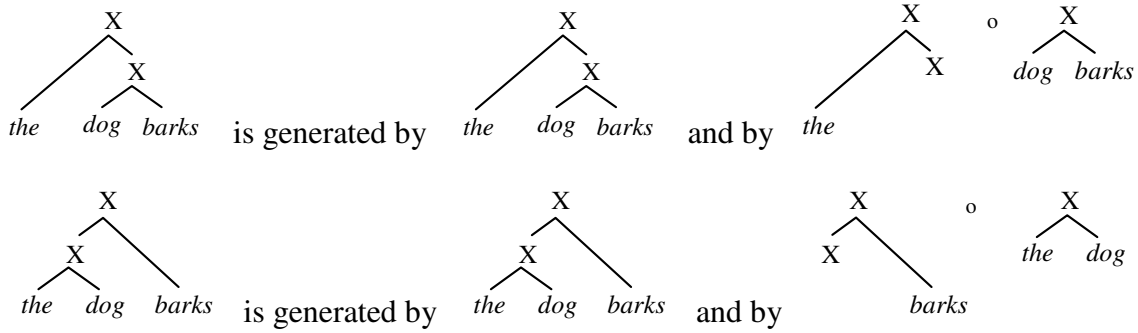


Figure 8. Parsing *the dog barks* from the subtrees in Figure 7

Thus the shortest derivation is not unique: the sentence *the dog barks* can be trivially parsed by any of its fully spanning trees, which is a direct consequence of U-DOP's property that subtrees of any size may play a role in language learning. This situation does not usually occur when structures for *new* sentences are learned. For example, the shortest derivation for the new 'sentence' *watch dog barks* (using subtrees from Figure 7) is unique and given in Figure 9.

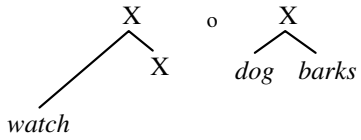


Figure 9. Unique shortest derivation for *watch dog barks* from the subtrees in Figure 7

But to decide between the trees in Figure 8 we need the subtree frequencies to break ties, that is, U-DOP computes the most probable tree from among the trees produced by the shortest derivations of *the dog barks*. The probability of a tree is computed from the frequencies of its subtrees in the same way as in the supervised version of DOP. Since the subtree [*the dog*] is the only subtree that occurs more than once, we can predict that the most probable tree corresponds to the structure [[*the dog*] *barks*] in Figure 7 where *the dog* is a constituent. This can also be shown formally by applying the probability definitions given in Section 2.

Thus the probability of the tree structure [*the* [*dog barks*]], is equal to the sum of the probabilities of its derivations in Figure 8. The probability of the first derivation consisting of the fully spanning tree is simply equal to the probability of selecting this tree from the space of all subtrees in Figure 7, which is 1/12. The probability of the second derivation of [*the* [*dog barks*]] in Figure 8 is equal to the product of the probabilities of selecting the two subtrees which is $1/12 \times 1/12 = 1/144$. The total probability of the tree is the probability that it is generated by any of its derivations which is the sum of the probabilities of the derivations:

$$P([\textit{the}\ [\textit{dog barks}]]) = 1/12 + (1/12 \times 1/12) = 13/144.$$

Similarly, we can compute the probability of the alternative tree structure, [[*the dog*] *barks*], which follows from its derivations in Figure 8. Note that the only difference is the probability of the subtree [*the dog*] being 2/12 (as it occurs twice). The total probability of this tree structure is:

$$P([[\textit{the dog}]\ \textit{barks}]) = 1/12 + (1/12 \times 2/12) = 14/144.$$

Thus the second tree wins, although with just a little bit. We leave the computation of the conditional probabilities of each tree given the sentence *the dog barks* to the reader (these are computed as the probability of each tree divided by the sum of probabilities of all trees for *the dog barks*). The relative difference in probability is small because the derivation consisting of the entire tree takes a considerable part of the probability mass (1/12). This simple example is only intended to illustrate U-DOP's probability model. In our experiments we will be mostly interested in learning structures for *new* sentences, where it is not the case that every sentence can be parsed by all fully spanning trees, as occurred with the example *watch dog barks* in Figure 9 which leads to a unique shortest derivation of largest possible chunks from the corpus.

For the sake of simplicity, we only used trees without lexical categories. But it is straightforward to assign abstract labels *X* to the words as well. If we do so for the sentences in Figure 6, then one of the possible subtrees for the sentence *watch the dog* is given in Figure 10. This subtree has a discontiguous yield *watch X dog*, which we will therefore refer to as a *discontiguous subtree*.

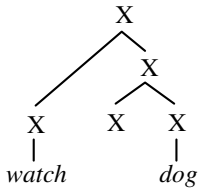


Figure 10. A discontiguous subtree

Discontiguous subtrees are important for covering a range of linguistic constructions, as those given in italics in sentences (1)-(6):

(1) BA carried *more* people *than* cargo in 2005.
(2) *What's* this scratch *doing* on the table?
(3) Don't *take* him *by surprise*.
(4) Fraser *put* dollie nighty *on*.
(5) Most software *companies* in Vietnam *are* small sized.

These constructions have been discussed at various places in the literature (e.g. Bod 1998, Goldberg 2006), and all of them are discontiguous. They range from idiomatic, multi-word units (e.g. (1)-(3)) and particle verbs (e.g. (4)) to regular syntactic phenomena as in (5). The notion of subtree can easily capture the syntactic structure of these discontiguous constructions. For example, the construction *more ... than …* in (1) may be represented by the subtree in Figure 11.

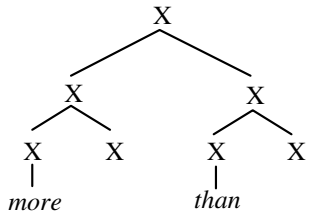


Figure 11. Discontiguous subtree for the construction *more...than...*

In our experiments in the following sections we will isolate the contribution of non-adjacent dependencies in learning the correct structures of utterances as well as in learning syntactic facets such as auxiliary fronting.

We should stress that the illustration of U-DOP's (and DOP's) working above has been mainly conceptual: in practice we do not compute the MPSD by first extracting all subtrees but by using a compact reduction of DOP proposed in Goodman (1996, 2003). This reduction is explained in the Appendix at the end of this paper and reduces the exponentially large number of corpus-subtrees to exactly 8 indexed 'PCFG' (Probabilistic Context-Free Grammar) rules for each internal node in a corpus-tree. This set of indexed PCFG rules generates the same derivations with the same probabilities as DOP and U-DOP and is therefore said to be isomorphic to (U-)DOP

(even though the term 'PCFG' is not entirely correct since the set of 'indexed PCFG rules' does not correspond to a standard PCFG in the literature – see the Appendix). The importance of the PCFG reduction method can hardly be overestimated, as may be illustrated by the combinatorial explosion of subtrees before applying the reduction method. For example, for the WSJ10 corpus of 7422 sentences no longer than 10 words, the number of subtrees assigned by U-DOP corresponds to almost 500 million while the number of indexed rules in the PCFG reduction is "only" 328 thousand (which is not a particularly large number in the current parsing systems – see Section 4). For conceptual reasons, we will often talk about 'subtrees' rather than 'indexed PCFG rules' as long as no confusion arises.

4 Experiments with Adult language

The illustration of U-DOP in the previous section was mainly based on artificial examples. How well does U-DOP learn constituent structures for sentences from more realistic settings? In this section we will carry out a (corpus-based) experiment with adult language, after which we will extend our experiments to child language in the following section. The main reason to test U-DOP on adult language is that it allows for comparing the model against a state-of-the-art approach to structure induction (Klein and Manning 2004). Only in Sections 5 and 6 will we investigate U-DOP's capacity to learn specific syntactic facets such as particle verbs and auxiliary inversion.

4.1 Experiments with the Penn, Negra and Chinese treebank

The Penn treebank (Marcus et al. 1993) has become a gold standard in evaluating natural language processing systems (see Manning and Schütze 1999) and has also been employed in linguistic research (Pullum and Scholz 2002). More recently, the Penn treebank has been used to evaluate *unsupervised* language learning models as well. Early approaches by van Zaanen (2000) and Clark (2001) tested on Penn's ATIS corpus, as did Solan et al. (2005), while Klein and Manning (2002, 2004, 2005) tested their systems on the larger Wall Street Journal corpus in the Penn treebank, as well as on the Chinese Treebank and the German Negra corpus. While these corpora are limited to specific domains of adult language use, it has been argued that relative frequencies of words, phrases etc. are rather stable across different domains (see Clark 2005).

U-DOP distinguishes itself from other learning models by its agnostic approach: All subtrees, be they contiguous or discontiguous, may contribute to learning the correct constituent structures. This is different from other learning approaches, including the well-known Constituent-Context Model (CCM) by Klein and Manning (2002, 2005). While CCM takes into account "all contiguous subsequences of a sentence" (Klein and Manning 2005: 1410), it neglects dependencies that are *non*-contiguous such as between *closest* and *to* in *the closest station to Union Square*. Moreover, by learning from linear subsequences only, CCM may underrepresent *structural* context. It is therefore interesting to experimentally compare U-DOP to these approaches and to assess whether there is any quantitative contribution of U-DOP's discontiguous subtrees.

As a first test, we evaluated U-DOP on the same data as Klein and Manning (2002, 2004, 2005): the Penn treebank WSJ10 corpus, containing human-annotated phrase-structure trees for 7422 sentences ≤ 10 words after removing punctuation, the German NEGRA10 corpus (Skut et al. 1997) and the Chinese CTB10 treebank (Xue et al. 2002) both containing annotated tree structures for 2200+ sentences ≤ 10 words after removing punctuation. As with most other unsupervised parsing models, we train and test on word strings that are already enriched with the Penn treebank

part-of-speech sequences rather than on word sequences directly. The actual goal is of course to directly test on word sequences, which will be carried out in the following sections.

For example, the word string *Investors suffered heavy losses* is annotated with the part-of-speech string NNS VBD JJ NNS, and is next assigned a total of five binary trees by U-DOP, listed in Figure 12 (where NNS stands for plural noun, VBD for past tense verb, and JJ for adjective).

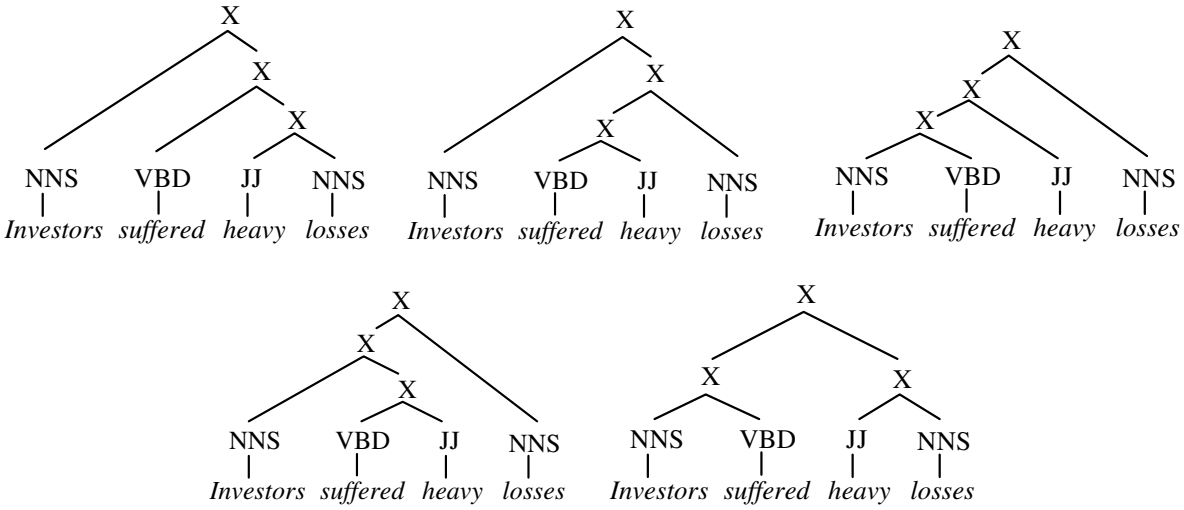


Figure 12. All binary trees for the WSJ sentence *Investors suffered heavy losses*

We used Goodman's reduction method to convert the set of U-DOP's trees (and subtrees) into a compact set of indexed PCFG rules (see Appendix). For the 7422 sentences from the WSJ10 corpus, this resulted in 328,018 different indexed PCFG-rules. This number is not exceptionally large in the field of natural language processing: Current parsing models often use more than one million rules (Collins and Duffy 2002; Bod 2003) or even several millions of rules (Chiang 2007).

We will use the same evaluation metrics as Klein and Manning (2002, 2004), i.e. 'unlabeled precision' (UP) and 'unlabeled recall' (UR). These metrics compute respectively the percentage of correctly predicted constituents with respect to all constituents predicted by the model (UP), and the percentage of correctly predicted constituents with respect to the constituents in the treebank (UR). The two metrics of UP and UR are combined by the f-score F1 which is the harmonic mean of UP and UR: F1 = 2*UP*UR/(UP+UR). It should be kept in mind that this evaluation metric is taken from the evaluation procedures of *supervised* parsing systems which aim at mimicking the treebank annotations. Since the trees in the Penn treebank are quite shallow, this evaluation metric punishes systems that learn binary trees. Therefore, the treebank trees are (automatically) binarized in the same way as Klein and Manning (2002, 2004). For our first experiment we test on the full corpora, just as in Klein and Manning's work, after which we will employ *n*-fold cross-validation.

Tabel 1 shows the unlabeled precision (UP), unlabeled recall (UR) and the f-scores (F1, given in bold) of U-DOP against the scores of the CCM model in Klein and Manning (2002), the dependency learning model DMV in Klein and Manning (2004) as well as their combined model DMV+CCM which is based on both constituency and dependency. The table also includes a previous experiment with U-DOP in Bod (2006b), which we refer to as U-DOP'2006, where only a random sample of the subtrees was used.

Model	English (WSJ10)			German (NEGRA10)			Chinese (CTB10)		
	UP	UR	**F1**	UP	UR	**F1**	UP	UR	**F1**
CCM	64.2	81.6	**71.9**	48.1	85.5	**61.6**	34.6	64.3	**45.0**
DMV	46.6	59.2	**52.1**	38.4	69.5	**49.5**	35.9	66.7	**46.7**
DMV+CCM	69.3	88.0	**77.6**	49.6	89.7	**63.9**	33.3	62.0	**43.3**
U-DOP'2006	70.8	88.2	**78.5**	51.2	90.5	**65.4**	36.3	64.9	**46.6**
U-DOP	75.9	90.9	**82.7**	52.4	91.0	**66.5**	37.6	65.7	**47.8**

Table 1. Unlabeled Precision, Unlabeled Recall and F1-scores of U-DOP tested on the full English WSJ10, German NEGRA10 and Chinese CTB10, compared to other models.

The table indicates that U-DOP obtains competitive results compared to Klein and Manning's models, for all three metrics. The relatively high scores of U-DOP may be explained by the fact that the model takes into account (also) non-contiguous context in learning trees. We will investigate this hypothesis below. Note that the precision and recall scores differ substantially, especially for German. While most models obtain good recall scores (except for Chinese), the precision scores are disappointingly low. The table also shows that U-DOP's use of the entire subtree-set outperforms the experiment in Bod (2006b) where only a sample of the subtrees was used. More subtrees apparently lead to better predictions for the correct trees (we will come back to this in more detail in Section 5.3). Note that the scores for German and Chinese are lower than for English; we should keep in mind that the WSJ10 corpus is almost four times as large as the NEGRA10 and CTB10 corpora. It would be interesting to study the effect of reducing the size of the WSJ10 to roughly the same size as NEGRA10 and CTB10. We therefore carried out the same experiment on a smaller, random selection of 2200 WSJ10 sentences. On this selection, U-DOP obtained an f-score of 68.2%, which is comparable to the f-score on German sentences (66.5%) but still higher than the f-score on Chinese sentences (47.8%). This result is to some extent consonant with work in supervised parsing of Chinese which generally obtains lower results than parsing English (cf. Hearne and Way 2004).

We now come to isolating the effect of non-linear context in structure learning, as encoded by discontiguous subtrees, a feature which is not in the models of Klein and Manning. In order to test for statistical significance, we divide each of the three corpora into 10 training/test set splits where each training set constitutes 90% of the data and each test set 10% of the data (10-fold cross-validation). The strings in each training set were assigned all possible binary trees that were employed by U-DOP to compute the best tree for each string from the corresponding test set. For each of the 10 splits, we performed two experiments: one with all subtrees and one without discontiguous subtrees – or isomorphic PCFG-reductions thereof (Goodman 2003, p. 134, showed that his reduction method can just as well be applied to restricted subtree sets rather than DOP's full subtree set – see the Appendix). In Figure 13, subtree (a) is discontiguous, while the other two subtrees (b) and (c) are contiguous.

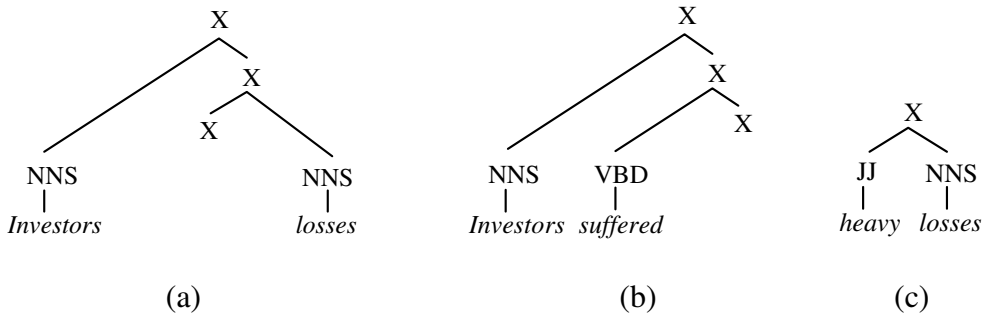


Figure 13. One discontiguous subtree and two contiguous subtrees from Figure 12

Table 2 shows the results of these experiments, where we focus on the average f-scores of U-DOP using all subtrees and of U-DOP using only contiguous subtrees. We also added the f-scores obtained by Klein and Manning's CCM, DMV and DMV+CCM models that were tested on the entire corpora.

Model	**English** (WSJ10)	**German** (NEGRA10)	**Chinese** (CTB10)
U-DOP With All Subtrees	80.3	64.8	46.1
U-DOP Without Discontiguous Subtrees	72.1	60.3	43.5
CCM	71.9	61.6	45.0
DMV	52.1	49.5	46.7
DMV+CCM	77.6	63.9	43.3

Table 2. F-scores of U-DOP for WSJ10 with and without discontiguous subtrees using 10-fold cross-validation

As seen in the table, the full U-DOP model scores consistently better than the U-DOP model without discontiguous information.[4] All differences in f-scores were statistically significant according to paired *t*-testing ($p<0.02$ or smaller). The f-scores of Klein and Manning's models are only added for completeness, since they were obtained on the entire corpus, rather than on 10 splits. Although exact comparison is not possible, it is interesting that without discontiguous subtrees U-DOP obtains results that are similar to the CCM model which is based on contiguous dependencies only. In any case, our experiments show that discontiguous dependencies contribute to significantly higher f-score in predicting the correct trees. This result is consonant with U-DOP's cognitive claim that all possible subtrees should be taken into account, or in other words, that no structural-lexical relation should be neglected in learning the syntactic analyses of sentences. We will go into a more qualitative analysis of discontiguous subtrees in the following sections. The *coverage* (i.e. the percentage of sentences that could be parsed) by U-DOP was 100% for all training/test set splits. This is not surprising, because every label *X* can be substituted into every other label *X* in U-DOP. The actual challenge is to find the best structure for a sentence.

We should keep in mind that all experiments so far have been carried out with tagged sentences. Children do not learn language from sentences enriched with part-of-speech categories, and if we want to investigate the cognitive plausibility of U-DOP we need to apply the model

[4] Note that the best f-scores in Table 2 are somewhat lower than U-DOP's f-scores in Table 1. This is due to testing on smaller parts of the corpora (*n*-fold testing) rather than testing on the full corpora.

directly to word strings from child language, which we will do so in Section 5. For completeness we mention that an experiment with U-DOP on WSJ10 *word* strings yielded only 51.7% f-score. By adding an unsupervised part-of-speech tagger based on distributional clustering (Clark 2000), we obtain an f-score of 76.4%, which is just 6% lower than by testing on the WSJ10 part-of-speech sequences. It would be interesting to generalize unsupervised part-of-speech tagging to German and Chinese, and test U-DOP on these data as well, but this falls beyond the scope of this paper.

4.2 The problem of 'distituents'

There is an important question as to whether U-DOP does not overlearn highly frequent word combinations that are non-constituents, also known as 'distituents'. For example, word combinations consisting of a preposition followed by a determiner, such as *in the*, *on the*, *at the* etc., occur in the top four most frequent co-occurences in the Wall Street Journal, and yet they do not form a constituent. The constituent boundary always lies between the preposition and the determiner, as in [*in* [*the city*]], which in the Penn treebank part-of-speech notation corresponds to [IN [DT NN]]. There are many types of combinations that are far less frequent than IN DT and that do form constituents. How does U-DOP deal with this?

Let's have a look at the most frequent constituent types learned by U-DOP in our experiments on the WSJ10 (Table 1) and compare them with the most frequent substrings from the same corpus. As in Klein and Manning (2002), we mean by a constituent type a part-of-speech sequence that constitutes a yield (i.e. a sequence of leaves) of a subtree in the best tree. Table 3 shows the 10 most frequently induced constituent types by U-DOP together with the 10 actually most frequently occurring constituent types in the WSJ10, and the 10 most frequently occurring part-of-speech sequences (which turn out all to be bigrams). We thus represent the constituent types by their corresponding lexical categories. For instance, DT NN in the first column refers to a determiner-(singular)noun pair, while DT JJ NN refers to determiner-adjective-(singular)noun triple.[5]

[5] The full list of lexical categories in the Penn Treebank II (Marcus et al. 1993) are: CC - Coordinating conjunction; CD - Cardinal number; DT - Determiner; EX - Existential there; FW - Foreign word; IN - Preposition or subordinating conjunction; JJ – Adjective; JJR - Adjective, comparative; JJS - Adjective, superlative; LS - List item marker; MD – Modal; NN - Noun, singular or mass; NNS - Noun, plural; NNP - Proper noun, singular; NNPS - Proper noun, plural; PDT – Predeterminer; POS - Possessive ending; PRP - Personal pronoun; PRP$ - Possessive pronoun; RB – Adverb; RBR - Adverb, comparative; RBS - Adverb, superlative; RP – Particle; SYM – Symbol; TO – to; UH – Interjection; VB - Verb, base form; VBD - Verb, past tense; VBG - Verb, gerund or present participle; VBN - Verb, past participle; VBP - Verb, non-3rd person singular present; VBZ - Verb, 3rd person singular present; WDT - Wh-determiner; WP - Wh-pronoun; WP$ - Possessive wh-pronoun; WRB - Wh-adverb.

Rank	Most frequent U-DOP constituents	Most frequent WSJ10 constituents	Most frequent WSJ10 substrings
1	DT NN	DT NN	NNP NNP
2	NNP NNP	NNP NNP	DT NN
3	DT JJ NN	CD CD	JJ NN
4	IN DT NN	JJ NNS	IN DT
5	CD CD	DT JJ NN	NN IN
6	DT NNS	DT NNS	DT JJ
7	JJ NNS	JJ NN	JJ NNS
8	JJ NN	CD NN	NN NN
9	VBN IN	IN NN	CD CD
10	VBD NNS	IN DT NN	NN VBZ

Table 3. Most frequently learned constituent types by U-DOP for WSJ10, compared with most frequently occurring constituent types in Penn treebank WSJ10, and the most frequently occurring part-of-speech sequences in Penn treebank WSJ10

In the table we see that a distituent type like IN DT (*in the*, *on the*, *at the* etc.) occurs indeed very frequently as a substring in the WSJ10 (third column), but not among U-DOP's induced constituents in the first column, and neither among the hand-annotated constituents in the middle column. Why is this? First note that there is another substring DT NN which occurs even more frequently than the substring IN DT (see third column of Table 3). U-DOP's probability model will then favor a covering subtree for IN DT NN which consists of a division into IN *X* and DT NN rather than into IN DT and *X* NN. As a consequence IN DT will not be assigned a constituent in the most probable tree. The same kind of reasoning can be made for a subtree for DT JJ NN where the constituent JJ NN occurs more frequently as a substring than the distituent DT JJ. In other words: while distituents like IN DT and DT JJ occur in the top most frequent part-of-speech strings, they are not learned as constituents by U-DOP's probability model. This shows that the influence of frequency is more subtle than often assumed. For example, in Bybee and Hopper (2001:14) we read that "Constituent structure is determined by frequency of co-occurrence [...]: the more often two elements occur in sequence the tighter will be their constituent structure"). This idea, as attractive as it is, is incorrect. It is not the simple frequency of co-occurrence that determines constituent learning, but the *probability of the structure* of that co-occurrence. (This is not to say that a collocation of the form IN DT cannot form a phonetic phrase. What we have shown is that the learning of *syntactic* phrases, such as noun phrase and prepositional phrase, is more complex than applying simple frequency.)

5 Experiments with the Childes database

To test U-DOP on child language, we used the Eve corpus (Brown 1973) in the Childes database (MacWhinney 2000). Our choice was motivated by the accurate syntactic annotations that have recently been released for this corpus (Sagae et al. 2007), as well as its central role in child language acquisition research (cf. Moerk 1983). The Eve corpus consists of 20 chronologically ordered files each of about 1-1.5 hour dialog between child and adult that cover the period of Eve's language development from age 1;6 till age 2;3 (with two-week intervals). During this period, Eve's language changes from two-word utterances like *More cookie* and *Papa tray* to

relatively long sentences like *Someone's in kitchen with Dinah* and *I made my baby sit in high chair*. In our learning experiments in this section we only use the files that were manually annotated and checked, which correspond to the first 15 files of the Eve corpus covering the age span 1;6-2;1 (but note that in section 6 we will use all files in our generation experiments). The hand-annotations contain dependency structures for a total of 65,363 words. Sagae et al. (2007) labeled the dependencies by 37 distinct grammatical categories, and used the part-of-speech categories as described in MacWhinney (2000). Of course there is a question whether the same categories can be applied to different stages of child language development. But since we will discard the category labels in our unlabeled trees in our evaluations (as U-DOP does not learn categories), we will not go into this question for the moment. We will see that unlabeled tree structures are expressive enough to distinguish, for example, between holophrases (as represented by fully lexicalized subtrees) and constructions with open slots (as represented by partially lexicalized subtrees).

The annotations in Sagae et al (2007) were automatically converted to unlabeled binary constituent structures using standard techniques (Xia and Palmer 2001). Arities larger than 2 were converted into binary right-branching such that we obtained a unique binary tree for each dependency structure. This resulted in a test corpus of 18,863 fully hand-annotated, manually checked utterances, 10,280 adult and 8,563 child. For example, the binary tree structure for the Eve sentence (from file 15) *I can blow it up* is given in Figure 14.

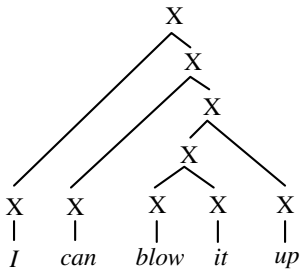


Figure 14. Binary tree for Childes sentence *I can blow it up* (Eve corpus)

5.1 Learning structures for the Eve corpus by U-DOP

Our main goal is to investigate in how far U-DOP can be used to incrementally model child language development. But as a baseline we first evaluate the *non*-incremental U-DOP model on the Eve corpus. We applied U-DOP to the word strings from the 15 annotated Eve files, where we distinguished between two subcorpora: Child (8,563 utterances) and Adult (10,280 utterances). As before, we used a PCFG reduction of U-DOP which resulted in a total of 498,826 indexed PCFG rules (remember that the number of subtrees and indexed PCFG rules increases when we add abstract labels to the words, as done for the Eve corpus).

As a first experiment we wanted to test in how far U-DOP could learn structures for the Child utterances on the basis of the Adult utterances only. This can be carried out by assigning all binary trees to the Adult utterances by which the best structures for Child utterances were computed (after which the outcome was evaluated against the hand annotations). However, for comparison we also lumped the Adult and Child utterances together as input, and used respectively the Child's structures as output. Additionally, we also carried out an experiment where the Child utterances are used as input and the Child structures as output (even though a child does of course not learn the structures entirely from its own language). Next, we did the

same for Adult utterances as input and Adult structures as output. Finally, to make our first set of experiments 'complete' we used Child utterances as input and Adult structures as output.

We should keep in mind that we did not use any part-of-speech annotations from the Eve corpus: we directly learned structures for *word* strings. This leads to the problem of unknown words, especially for the experiment from Child to Adult. As in Bod (1998, 2003), we assigned wildcards to unknown words such that they could match with any known word. Table 4 shows the results (unlabeled precision, unlabeled recall and f-score) where we again distinguish between using all subtrees and only contiguous subtrees. To the best of our knowledge these are the first published results on unsupervised structure induction for Childes data evaluated against the hand-assigned structures by Sagae et al. (2007).

Experimental Setting	**Full U-DOP**			**U-DOP without discontiguous subtrees**		
	UP	UR	**F1**	UP	UR	**F1**
Adult to Child	77.0	87.2	**81.8**	75.2	85.9	**80.2**
Child to Adult	45.5	51.4	**48.3**	44.1	50.3	**47.0**
Full corpus to Child	85.8	92.7	**89.1**	84.7	91.4	**87.9**
Full corpus to Adult	79.6	89.6	**84.3**	79.0	88.3	**83.4**
Child to Child	86.6	91.3	**88.9**	85.2	90.8	**87.9**
Adult to Adult	79.7	89.9	**84.5**	78.4	89.5	**83.6**

Table 4. Unlabeled Precision, Unlabeled Recall and F1-scores of U-DOP against hand-annotated Eve data in the Childes, under different experimental settings

The first thing that strikes us in Table 4 is the relatively low f-score for Child to Adult (48.3%). This low score is actually not surprising since the lexicon, as well as the grammar, of an adult are much larger than those of a child, which makes it hard to learn to parse adult sentences from child utterances only. Even when we discard all Adult sentences that have unknown words in the Child data, we still obtain an f-score of just 58.0%. What is more interesting, is that the cognitively more relevant experimental setting, Adult to Child, obtains a relatively high f-score of 81.8%. While this f-score is lower than Child to Child (88.9%), and the differences were statistically significant according to 10-fold cross-validation ($p < 0.01$), it is of course harder for U-DOP to learn the Child structures from Adult utterances, than it is to learn the Child structures from the child's own utterances. Yet, children do not learn a language by just listening to their own sentences, thus the Adult to Child setting is more relevant to the goal of modeling language learning. On the other hand, we should not rule out the possibility that children's utterances have an effect on their own learning. This is reflected by the Full corpus to Child setting, which obtains slightly better results than the Child to Child setting (the differences were not statistically significant according to 10-fold cross-validation) but it definitely obtained better results than Adult to Child (for which the differences were statistically significant, $p < 0.01$).

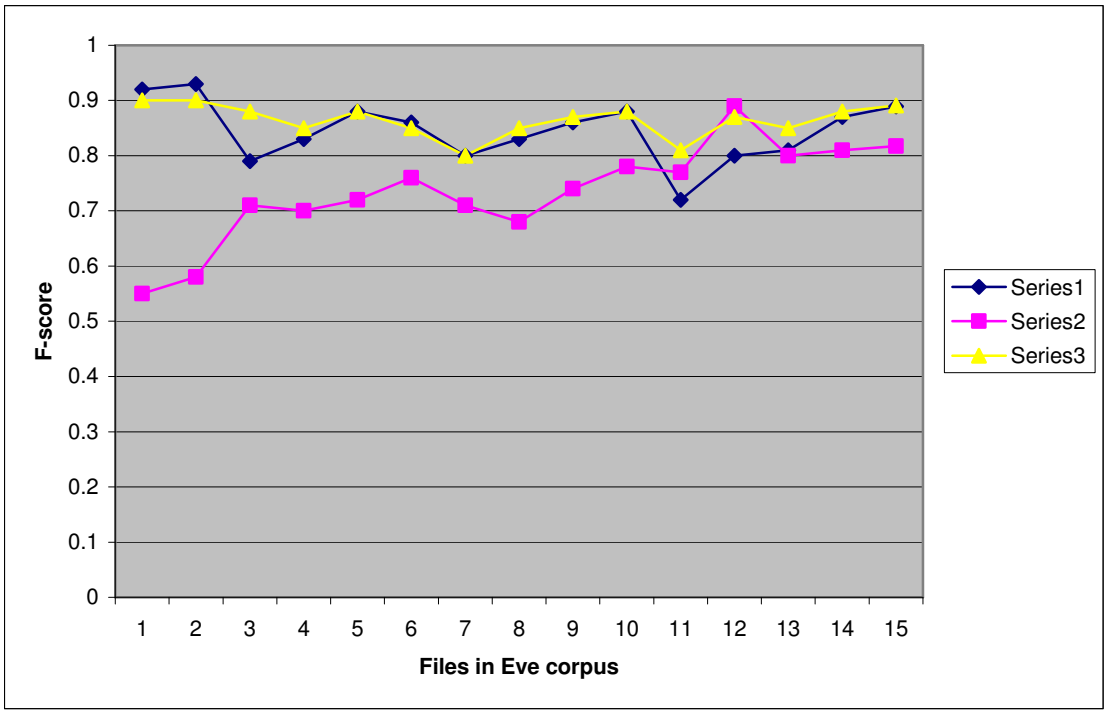
Table 4 shows that the use of all subtrees consistently outperforms the use of only contiguous subtrees. This is consonant with our results in the previous section. An additional experiment with 10-fold testing showed that the differences in f-score between full U-DOP and U-DOP without discontiguous subtrees are statistically significant for all data ($p < 0.05$ or smaller).[6]

[6] For reasons of completeness we have also evaluated several other versions of U-DOP. By testing U-DOP by means

Note that the precision and recall scores for each setting differ much less than in the experiments on adult language in Section 4 (Table 1), which can be considered an improvement with respect to the adult language experiments.

5.2 Extending U-DOP towards incremental learning

We will now extend U-DOP towards incremental learning by inducing the structures for the child utterances of each Eve file on the basis of the accumulated files of previous utterances up to the particular file. We took (respectively) the total Child utterances up to a certain file, the total Adult utterances up to certain file, and the total Child and Adult utterances taken together (i.e. what we called Full corpus above) in order to derive the structures for Eve for a particular (*non*-accumulated) file – which had the same file number as the last file of the accumulated files. In this way we create a first extension of U-DOP towards incremental learning: each file in the Eve corpus corresponds to a certain stage in Eve's language development, and we want to figure out in how far the structures for Eve can be derived from the accumulated language experiences (of Child, Adult and Full corpus) at each stage. Figure 15 gives the f-scores for each file where we distinguish between Accumulated Child to Child (Series 1), Accumulated Adult to Child (Series 2), and Accumulated Child and Adult to Child (Series 3). Remember that file 1 corresponds to age 1;6 and file 15 to age 2;1, with 2 week intervals between consecutive files.



of the most probable tree only (i.e. without the shortest derivation), significantly lower f-scores were obtained both with and without discontiguous subtrees. It thus seems to be important to *first compute the distribution of structurally most analogous trees, after which the statistics is applied*. To investigate other notions of "distribution of most analogous trees", we have tested also varieties of U-DOP by using the *k*-shortest (second shortest, third shortest etc.) derivations instead of the shortest derivation alone. While such an approach slightly improved the f-score for supervised DOP (cf. Bod 2002), it significantly deteriorated the f-scores for U-DOP.

Figure 15. F-scores for the Eve corpus, where U-DOP is tested on *Accumulated Child to Child* (Series1), *Accumulated Adult to Child* (Series2), and *Accumulated Child and Adult to Child* (Series3).

Figure 15 supports the observation that Adult to Child (Series2) is mostly harder than Child to Child (Series1), except for files 11 and 12 where Adult to Child outperforms Child to Child (remember that the f-scores are only computed on the most recent non-accumulated Child file). We do not know why this is; a closer look at the utterances gave no more hints than that Eve uses for the first time gerunds (e.g. *Sue giving some milk*), which occurred earlier and more frequently in Adult data than in Child data, and were parsed correctly only in the Adult to Child setting. Note that the Full corpus to Child (Series3) results obtains the best f-score in most cases. However, only for the Adult to Child setting there is a global increas in f-score from file 1 to file 15, while for the other two settings there is no improvement from file 1 to file 15. This is perhaps not surprising since the structure induction for Eve is accomplished in these settings by using Eve's utterances themselves as well (which is not the case in Adult to Child setting where U-DOP induced Eve's structures by means of the Adult utterances alone).

It may also be interesting to have a closer look at what happens with the f-score at file 3: Child to Child decreases while Adult to Child increases. This may indicate that there are new syntactic constructs that appear in file 3 which were not (yet) in the Child data but already available in the Adult data. A qualitative comparison between files 2 (age 1;6) and 3 (age 1;7) seemed to support this. For example, in file 3 Eve uses full-fledged sentences with the auxiliary *is* as in *the dog is stuck* (which previously would only occur as *dog stuck*). Moreover, Eve uses in file 3 for the first time verb combinations like *'d help* in *I'd help stool away*. These kind of constructions are very hard to process without examples from Adults.

Although an incremental model is cognitively more realistic than a non-incremental model, we should keep in mind that the data in the Eve corpus is not dense enough to model each step in Eve's language development. Yet, this data sparseness may perhaps be overcome if we apply U-DOP within one and the same Eve-file. In this way, we can test U-DOP's performance in a sentence-by-sentence way on Eve's data (rather than on a file-by-file way). One of the phenomena we have been interested in is the acquisition of discontiguous constructions, such as separable particle verbs. How does U-DOP simulate this learning process? To deal with this question, we applied the incremental version of U-DOP (using the Adult and Child to Child setting) to a sequence of Eve's utterances with the separable particle verb *blow … up* from file 15. Figure 16 lists these utterances with *blow … up*.

```
1. *CHI:  I trying a blow it up Fraser .
2. *CHI:  there I blow it up .
3. *CHI:  there I blow it up .
4. *CHI:  I can't .
5. *CHI:  there I blow it up .
6. *CHI:  I blow .
7. *CHI:  I have blow it up up big .
8. *CHI:  yeah .
9. *MOT:  you have to blow it up big ?
10.*MOT:  well I don't think you can Eve .
11.*MOT:  because there's knot in the balloon that I cannot get untied .
12.*MOT:  we'll have to get another one .
13.*CHI:  I can blow this up .
14.*MOT:  I don't think you can .
15.*CHI:  I can blow it in my mouth .
```

Figure 16. Dialog between Eve and her mother with the discontiguous phrase *blow … up*

Up to sentence 5 (in Figure 16), Eve seems to use the phrase *blow it up* as one unit in that there is no evidence for any internal structure of the phrase. U-DOP learned at sentence 2 that *blow it up* is a separate constituent, but was not able to induce any further internal structure for this constituent, and thus left open all possibilities (i.e. it maintained two different trees for *blow it up*). In sentence 6 Eve produces *I blow*, which led U-DOP to induce that *blow* is a separate constituent, but without being able to decide whether *it* is attached to *blow* or to *up*. The next major sentence is 13: *I can blow this up*. The new word *this* occurs between *blow* and *up* which led U-DOP to induce two possible subtrees: [[*blow X*] *up*] and [*blow* [*X up*]] without breaking ties yet. Finally, in sentence 15, Eve produces *blow it* without *up*, which led U-DOP to assign the subtree [[*blow X*] *up*] a higher frequency than [*blow* [*X up*]]. This means that (1) U-DOP has correctly learned the separable particle verb *blow … up* at this point, and (2) DOP's MPSD will block the production at this point of 'incorrect' constructions such as *blow up it* since only the larger (learned) construction will lead to the shortest derivation (we will extensively come back to generation in the next section).

A limitation of the experiment above may be that U-DOP could only learn the particle verb construction from the utterances produced by both her mother and by Eve herself (i.e. the Child and Adult to Child set-up). It would be interesting to explore whether U-DOP can also learn discontiguous phrasal verbs from adult utterances alone (i.e. Adult to Child set-up), such as the particle verb *put … in*, as shown in Figure 17.

```
1. *MOT:    well we can put it in .
2. *MOT:    yeah .
3. *MOT:    Mom can put the stick in .
4. *MOT:    we just can't put any air in .
```

Figure 17. Mother utterances from the Eve corpus with discontiguous phrase *put … in*

The four sentences in Figure 17 suffice for U-DOP to learn the construction *put X in*. At sentence 3, U-DOP induced that *can put it in* and *can put the stick in* are generalized by *can put X in*. But the internal structure remains unspecified. At sentence 4, U-DOP additionally derived that *put X in* can occur separately from *can*, resulting in an additional constituent boundary. Thus by initially

leaving open all possible structures, U-DOP incrementally rules out incorrect structures until the correct construction *put X in* is learned. In this example, U-DOP was not able to decide on any further internal structure for *put X in*, leaving open *all* (i.e. two) possibilities at this point. This is equivalent to saying that according to U-DOP *put X in* has *no* internal structure at this point.

Note that in both examples (i.e. *blow it up* and *put it in*), U-DOP follows a route from concrete constructions to more abstract constructions with open slots. The subtrees that partake in U-DOP's MPSD initially correspond to 'holophrases' after which they get more abstract resulting in the discontiguous phrasal verb. This is consonant with studies of child language acquisition (Peters 1983; Tomasello 2003) which indicate that children move from item-based constructions to contructions with open positions. Although this is an interesting result, we must keep in mind that Eve's files are separated by two-week time intervals during which there were important learning steps that have not been recorded and that can therefore not be modeled by U-DOP. Yet, we will see in Section 6 that the grammar underlying U-DOP's induced structures triggers some interesting new experiments regarding language generation.

5.3 The effect of subtree size

Before going into generation experiments with U-DOP/DOP, we want to test whether we can obtain the same (or perhaps better) f-scores by putting constraints on U-DOP. By limiting the *size* of U-DOP's subtrees we can instantiate various other models. We define the size of a subtree by its depth, which is the length of the longest path from root to leaf in a subtree. For example, by restricting the maximum depth of the subtrees to one, we obtain an unsupervised version of probabilistic context-free grammar or PCFG (such a PCFG should not be confused with a 'PCFG'-reduction of DOP's PTSG for which each node in the tree receives 8 indexed PCFG-rules, and which is not equal to the standard notion of a PCFG – see Appendix). When we allow subtrees of at most depth 2, we obtain an extension towards a lexicalized tree-substitution grammar. The larger the depth of the subtrees, and consequently the width, the more (sequential and structural) dependencies can be taken into account. But there is a question whether we need subtrees of arbitrary depth to get the highest f-score. In particular, do we need such large productive units for the earliest stages of Eve's language development? To test this, we split the hand-annotated part of the Eve corpus into three equal periods, each of which contains 5 files.

Table 5 shows the f-scores of U-DOP on the Adult to Child learning task for the three different periods with different maximum subtree depths. The average sentence length (a.s.l.) is also given for each period.

Maximum Subtree Depth	**File 1-5 a.s.l. = 1.84**	**File 6-10 a.s.l. = 2.59**	**File 11-15 a.s.l. = 3.01**
1 (PCFG)	49.5	44.2	35.6
2	76.2	64.0	57.9
3	88.7	78.6	68.1
4	88.6	80.5	74.0
5	88.7	84.9	75.8
6	88.6	84.9	76.3
All (U-DOP)	88.7	84.9	77.8

Table 5. F-scores of U-DOP on the Adult to Child learning task for three periods, where the subtrees are limited to a certain maximum depth. The a.s.l. refers to the average number of words per sentence (average sentence length). For all periods there are subtrees larger than depth 6.

The table shows that for the first period (file 1-5; age 1;6-1;8) the f-score increases up to subtree depth 3, while for the second period (age 1;8-1;10) the f-score increases up to subtree depth 5, and in the third period (age 1;11-2;1) there is a continuous increase in f-score with increasing subtree size. Thus the f-score decreases if the subtrees are limited to a simple PCFG, for all periods, and the subtree-depth for which maximum f-score is obtained increases with age (and corresponding average sentence length). This suggests that children's grammars move from small building blocks to grammars based on increasingly larger units. It is remarkable that the f-score continues to grow in the third period. We will study the qualitative effect of subtree-size in more detail in our generation experiment below.

6 Generation experiments with auxiliary fronting

So far we have shown how U-DOP can infer to some extent the syntactic structures of Child utterances from Adult utterances. But once we have learned these structures, we have also learned the grammar implicit in these structures by which we can generate new utterances, namely by combining subtrees from the learned structures. This DOP/PTSG model will of course overgenerate due to its lack of labels and absence of semantics. In principle, we need a DOP model that computes the best string for a given meaning representation, such as in Bod (1998). But in the absence of meaning in the current version of U-DOP, we can at least test whether the derived PTSG correctly generates certain syntactic facets of (child) language. In this section we will test our method on the phenomenon known as auxiliary fronting. We will deal with the phenomenon in two ways: first in a 'logical' way, similar to Clark and Eyraud (2006); next, in an empirical way by using the induced structures from the Eve corpus.

The phenomenon of auxiliary fronting is often taken to support the well-known "Poverty of the Stimulus" argument and is called by Crain (1991) the "parade case of an innate constraint". Let's start with the typical examples which are the same as those used in Crain (1991), MacWhinney (2005), Clark and Eyraud (2006) and many others:

(5) The man is hungry

If we turn sentence (5) into a (polar) interrogative, the auxiliary *is* is fronted, resulting in sentence (6).

(6) Is the man hungry?

A language learner might derive from these two sentences that the first occurring auxiliary is fronted. However, when the sentence also contains a relative clause with an auxiliary *is*, it should not be the first occurrence of *is* that is fronted but the one in the main clause:

(7) The man who is eating is hungry

(8) Is the man who is eating hungry?

Many researchers have argued that there is no reason that children should favor the correct auxiliary fronting. Yet children do produce the correct sentences of the form (7) and rarely of the form (9) even if they have not heard the correct form before (Crain and Nakayama 1987).[9]

(9) *Is the man who eating is hungry?

According to the nativist view and the poverty of the stimulus argument, sentences of the type in (8) are so rare that children must have innately specified knowledge that allows them to learn this facet of language without ever having seen it (Crain and Nakayama 1987). On the other hand, it has been claimed that this type of sentence can be learned from experience (Lewis and Elman 2001; Reali and Christiansen 2005). We will not enter the controversy on this issue (see Pullum and Scholz 2002; Kam et al. 2005), but believe that both viewpoints overlook an alternative possibility, namely that auxiliary fronting needs neither be innate nor in the input data to be learned, but that its underlying rule may be an emergent property of a structure learning algorithm. We will demonstrate that by U-DOP's shortest derivation, the phenomenon of auxiliary fronting does not have to be in the input data and yet can be learned.

6.1 Learning auxiliary fronting from a constructed example

The learning of auxiliary fronting can proceed when we have induced tree structures for the following two sentences (we will generalize over these sentences in Section 6.2):

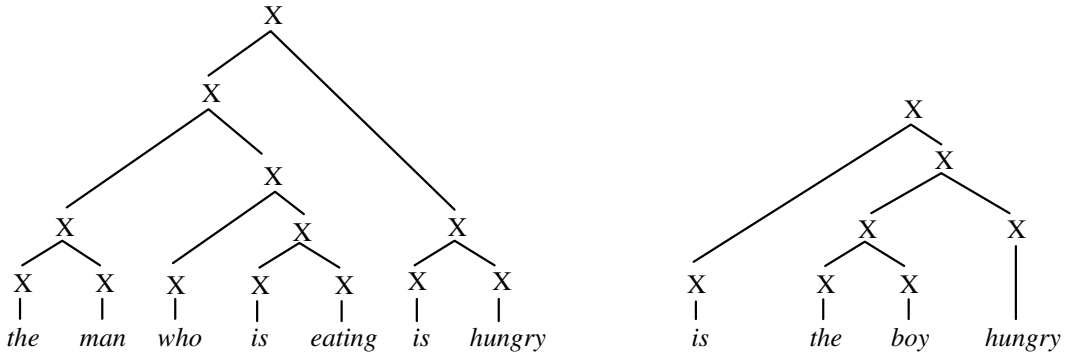
(10) The man who is eating is hungry
(11) Is the boy hungry?

Note that these sentences do not contain an example of complex fronting where the auxiliary should be fronted from the main clause rather than from the relative clause. The tree structures for (10) and (11) can be derived from exactly the same sentences as in Clark and Eyraud (2006):

(12) The man who is eating mumbled
(13) The man is hungry
(14) The man mumbled
(15) The boy is eating

The best trees for (10) and (11) computed by U-DOP from (10)-(15) are given in Figure 18.

[9] Crain and Nakayama (1987) found that children never produced the incorrect form (9). But in a more detailed experiment on eliciting auxiliary fronting questions from children, Ambridge et al. (2008) found that the correct form was produced 26.7% of the time, the incorrect form in (9) was produced 4.55% of the time, and auxiliary doubling errors were produced 14.02% of the time. The other produced questions corresponded to shorter forms of the questions, unclassified errors and other excluded responses.



(a) (b)

Figure 18. Tree structures for *the man who is eating is hungry* and *is the boy hungry?* learned by U-DOP from the sentences (10)-(15)

Given these trees, we can easily prove that the shortest derivation produces the correct auxiliary fronting. That is, in order to produce the correct AUX-question, *Is the man who is eating hungry*, we only need to combine the following two subtrees in Figure 19 from the acquired structures in Figure 18 (note that the first subtree is discontiguous)[9].

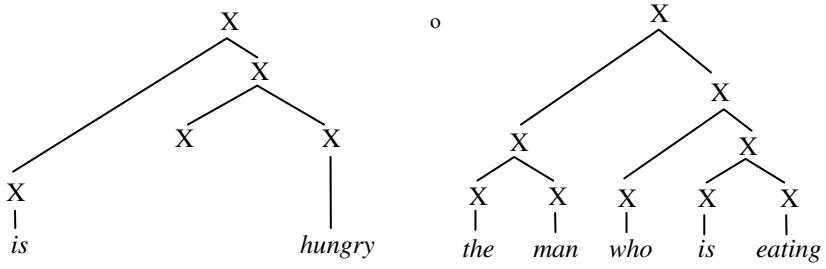


Figure 19. Producing the correct auxiliary fronting by combining two subtrees from Figure 18

Instead, to produce the incorrect AUX-question **Is the man who eating is hungry?* we would need to combine at least four subtrees from Figure 18 (which would in fact never be produced by the shortest derivation), which are given in Figure 20:

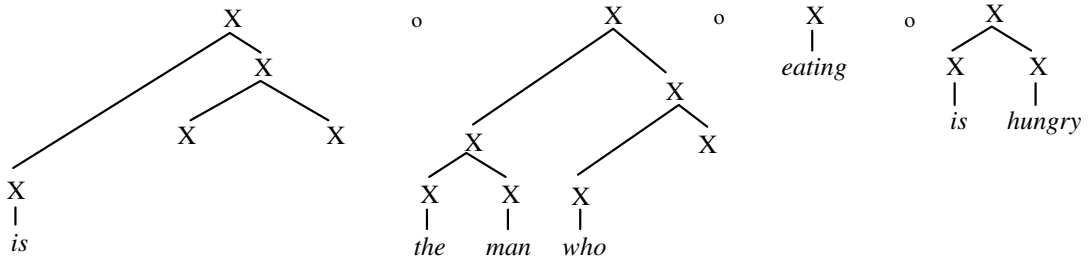


Figure 20. Producing the incorrect aux-fronting by combining four subtrees from Figure 18

[10] We are implicitly assuming a DOP model which computes the most probable shortest derivation given a certain meaning to be conveyed, such as in Bonnema et al. (1997) and Bod (1998).

Clearly the derivation in Figure 19 is the shortest one and produces the correct sentence, thereby overruling the incorrect form. Although our argument is based on one example only (which we will extend in Section 6.2), it suggests the following explanation for auxiliary fronting: the shortest derivation provides the maximal similarity or analogy between the new sentence (with complex fronting) and old sentences, that is, there is maximal structure sharing between new and old forms (cf. Gentner and Markman 1997). As an effect, the shortest derivation substitutes the simple NP *the boy* for the complex NP *the man who is eating*, leading to the correct fronting (see Bod 2007b).

The example above thus shows that **Is the man who eating is hungry?* is blocked by the MPSD, provided that we have sentences like (10)-(15) (which we will generalize to the entire Eve corpus in Section 6.2). But we have not yet shown that a sentence like **Is the man who is eating is hungry?* is also blocked. This incorrect sentence can in fact also be generated by only two subtrees from Figure 18 (i.e. by combining the subtree $[\text{is}_X\ X]_X$ from 18b and the entire tree from 18a) and would thus compete with the correct *Is the man who is eating hungry?* Interestingly, Crain and Nakayama (1987) report that children make the same type of error with auxiliary doubling (also discussed in Ambridge et al. 2008). Yet if we additionally take into account the frequencies of the subtrees, it turns out that the MPSD is unique and predicts the correct fronting. This can be informally seen as follows. Since we also heard a sentence like (12) above (*The man who is eating mumbled*), the total frequency of the subtree for *The man who is eating* is twice as high as the subtree for *The man who is eating is hungry*, which means that the sentence with the correct auxiliary placement will win in this specific case. And in the more general case, a subtree for a sentence like *The man who is eating X*, where *X* stands for another constituent will be more frequent than a (sub)tree for the specified sentence *The man who is eating is hungry* (because the former sentence can occur with different fillers for *X*). Our argument does therefore not hinge on the specific example in this section. Thus if we leave out frequency, the shortest derivation generates one other *incorrect* auxiliary fronting, which is however also produced by children in the Crain and Nakayama (1987) experiment. But when we take into account frequency, the correct fronting will get a higher probability than the incorrect fronting.

6.2 Learning auxiliary fronting from the Eve corpus

The example in the previous section is limited to just a couple of artificial sentences. There is an important question as to whether we can generalize our artificial result to actual data. So far we have only shown that U-DOP/DOP can infer a complex AUX-question from a simple AUX-question and a complex declarative. But a language learner does not need to hear each time a new pair of sentences to produce a new AUX-question -- such as *Is the girl alone?* and *The girl who is crying is alone* in order to produce *Is the girl who is crying alone?*. In the following we will investigate whether U-DOP can learn auxiluary fronting from the Eve utterances rather than from constructed examples, and whether the model can derive the abstract generalization for the phenomenon.

First note that the patterns of respectively a complex declarative and a simple question in (10) and (11) can also be represented by (16) and (17) (with the only difference that *P* in (10) refers to *the man* while in (11) it refers to *the boy*, but this does not change our argument).

(16) P who is Q is R
(17) is P R?

We will assume that the variables P, Q and R can be of any (lexical or syntactic) category, except the auxiliary *is.* This assumption can lead to the production of implausible and unacceptable sentences, but our first goal will be to test whether U-DOP can generate the correct pattern *Is P who is Q R?* from (16) and (17) -- and we will see below that the AUX-questions generated by DOP are mostly acceptable, due to its preference of using largest possible chunks. Our question is thus whether U-DOP can assign structures to (16) and (17) on the basis of the Eve corpus such that the complex pattern (18) is generated by the (most probable) shortest derivation while the patterns (19) and (20) are not. For convenience we will refer to pattern (19) as "incorrect auxiliary fronting" and to pattern (20) as "auxiliary doubling".

(18) Is P who is Q R?
(19) *Is P who Q is R?
(20) *Is P who is Q is R?

We should first mention that there is no occurrence of the complex AUX-question (18) in the Eve corpus. Thus U-DOP cannot learn the complex pattern by simply using a large subtree from a single sentence. Moreover, there is no occurrence of the complex declarative (16) (*P who is Q is R*) in the Eve corpus either (although there are many instances of simple polar interrogatives like (17) as well as many relative clauses with *who*). This means that we cannot show by our experiment that the complex AUX-question can be derived from an *observed* complex declarative and a simple AUX-question. But it is interesting to investigate whether we can derive the complex AUX-question from raw data by learning first the structure of a complex declarative. Such an experiment could connect our 'logical' argument in Section 6.1 with a more empirical argument. Thus we first tested whether the structures of (16) and (17) could be derived from the Eve corpus. We used U-DOP's inferred structures in the *Adult to Child* setting from Section 5 to compute the MPSD for the two patterns *P who is Q is R*.and *is P R?* where *P, Q* and *R* were taken as wildcards. By employing the Adult to Child setting, we only use the structures learned for Eve's utterances which means that our result does not depend on the structures of Adult utterances.

The induced structures by U-DOP's MPSD for (16) and (17) are given in (21) and (22). For readability we will leave out the labels *X* at the internal nodes (in the sequel we only show the label *X* if it appears at an external node, for example in a subtree-yield).

(21) [[[P] [who [is Q]]] [is R]]
(22) [is [P R]]?

Note that (21) and (22) are virtually equivalent to the structures 14(a) and 14(b) modulo the internal structure of *P* which in (21) and (22) is taken as a whole constituent. On the basis of these two structures, DOP will generate the correct AUX-question by the shortest derivation in the same way as shown in Section 6.1 (Figure 19), namely by combining the two subtrees *[is [X R]]* and *[[P] [who [is Q]]]*, while the sentence with incorrect auxiliary fronting can be generated only by (at least) four subtrees (and the same argument in Section 6.1 can also be used for the auxiliary doubling). While this empirical result thus generalizes over the artificial result above, our experiment is based on the assumption that the structures (21) and (22) are the only trees that contribute to generating the AUX-question *Is P who is Q R?*. This is an unrealistic assumption: there are many other utterances in the Eve corpus whose subtrees may contribute to generating AUX-questions.

In our next experiment we will therefore use U-DOP's induced structures for Eve's utterances to compute the most probable shortest derivations *directly* for the patterns (18)-(20), rather than via the complex declarative. Table 10 gives for each AUX-pattern the minimal number of subtrees from Eve's utterance-structures that generated it, and the probability of the most probable tree among the shortest derivations.

Pattern	**Length of shortest derivation**	**MPSD** (probability of best tree)
Is P who is Q R?	3	$4.4 \cdot 10^{-17}$
*Is P who Q is R?	3	$2.1 \cdot 10^{-18}$
*Is P who is Q is R?	3	$1.8 \cdot 10^{-18}$

Table 10. Patterns of auxiliary fronting together with the length of the shortest derivation and the probability of the MPSD, as generated by subtrees from the induced structures of Eve's data.

Different from the artificial example above, all patterns are now generated by three subtrees – both the correct, incorrect and auxiliary-doubling patterns (remember that the correct AUX-question cannot be generated anymore from the complex declarative, as the latter does not appear in the Eve corpus). Tabel 10 shows that the probability of the correct fronting pattern is one order of magnitude higher than the probabilities of the other two patterns. The incorrect fronting pattern is slightly more likely than the auxiliary doubling pattern, while the study by Ambridge et al (2008) shows that auxiliary doubling is actually generated three times more often by children than the incorrect fronting in eliciting complex AUX-questions (roughly 14% against 4.5%). Yet, our experiment is not directly comparable to Ambridge et al. (2008) because the children in Ambridge et al. are on average 3.5 years older than Eve. It would be interesting to know the kind of auxiliary fronting sentences elicited from children of Eve's age – if possible at all. In any case, our experiment correctly predicts that the correct fronting has the highest probability.

While this experiment demonstrates that on the basis of unsupervised learning the correct abstract 'rule'-pattern for auxiliary fronting obtains a higher probability than the incorrect 'rule'-patterns, we should keep in mind that it is a *parsing* experiment rather than a generation experiment: we have parsed pre-given patterns instead of generating them. Children do of course not produce sequences of words with open slots but sequences of consecutive words. In our third experiment we therefore want to randomly generate a large number of complex AUX-questions so as to determine the percentage of the different auxilary patterns produced by U-DOP's derived PTSG. Note that we cannot exhaustively generate all possible questions, since there are infinitely many of them. Even the generation of all possible AUX-questions of maximally 8 words from the Eve corpus already leads to an unmanageably large number of sentences. Thus we must somehow sample from the distribution of possible AUX-questions if we want to investigate the percentages of various AUX-questions produced by U-DOP/DOP. Since we know that the correct AUX-question can be generated by three subtrees, we will produce our random generations by selecting (randomly) three subtrees of the following types:

(1) a subtree with the word *is* at the leftmost terminal of the subtree-yield (without any other restrictions),
(2) a subtree with the word *who* at any position in the subtree-yield,
(3) a subtree with the word *is* at any position in the subtree-yield .

Next we combine these 3 subtrees in the order of being sampled (if they can be combined at all). If the resulting sentence has all slots filled with words, we accept it, otherwise we discard it. In this way, we effectively sample from the distribution of shortest derivations for sentences of a large variety of patterns, many of which may be 'unacceptable', but which include patterns (18)-(20). If more than one derivation for the same sentence was generated then their probabilities were added, so as to take into account the MPSD. A total of 10 million sentences were randomly generated in this way, of which 3,484 had all slots filled. These were automatically compared with the three patterns (18)-(20). Table 11 gives the percentage of these patterns, as well as the other patterns that resulted from the generation experiment.

Pattern	**Percentage**
Is P who is Q R?	40.5
*Is P who Q is R?	6.9
*Is P who is Q is R?	7.0
Other:	
*Is P Q who is R?	10.7
*Is P Q R who is?	6.7
*Is who is P Q R?	4.0
*Is who P is Q R?	3.8
*Is who P Q R is?	2.5
*Is P is who Q R?	2.0
Etc…	
Total Other:	45.6

Table 11. Percentage of generated AUX-patterns by random generation of derivations of three subtrees with the words *is*, *who*, and *is*.

Table 11 shows a distribution where the correct fronting pattern is most likely, while the incorrect fronting and the auxiliary doubling are again almost equally likely. Although the correct fronting occurs only 40.5% of the time, it corresponds to the MPSD. Almost half of the generated sentences (45.6%) did not correspond to one of the three original patterns. In particular, the pattern **Is P Q who is R?* was generated quite frequently (10.7%). This pattern was not investigated in detail in the study by Ambridge et al. (2008), although under "other excluded responses" in Appendix E of their paper they list several sentences that are very similar to this pattern (e.g. *Is the boy washing the elephant who's tired*). The other incorrect patterns in Table 11 are not reported in Ambridge et al. (2008). It has of course to be seen which of these incorrect patterns will still be generated if we extend U-DOP with category induction (as we discuss in Section 7). But it is promising that our results are more in line with the recent experiments by Ambridge et al., in which various incorrect auxiliary fronting errors are reported, than with the older study by Crain and Nakayama (1987), in which incorrect fronting was never generated by children.

If we have a look at the sentences corresponding to the *correct* AUX-fronting pattern *Is P who is Q R?*, then it is remarkable that many of them are syntactically well-formed, and some of them are semantically plausible, even though there were no restrictions on the lexical/syntactic categories. This may be due to U-DOP/DOP's use of large chunks that tend to maintain collocational relations. Table 12 gives the ten most frequently generated AUX-questions of the pattern *Is P who is Q R?*, together with their unlabeled bracketings and their frequencies of being

generated (as well as the percentage corresponding to this frequency in the class of correct AUX-questions). It turns out that these sentences have roughly the same structure as in Figure 18(a). Note that the most frequently generated sentences also seem to correspond to the syntactically most acceptable and semantically most plausible sentences.

AUX-questions of the pattern *Is P who is Q R?* with induced unlabeled bracketings	**Frequency of being generated**
[Is [[Fraser [who [is crying]]] going]]	37 (2.6%)
[Is [[Fraser [who [is that]]] [having coffee]]]	30 (2.1%)
[Is [[Fraser [who [is crying]]] [having coffee]]]	28 (2.0%)
[Is [[that [who [is crying]]] [some noodles]]]	27 (1.9%)
[Is [[that [who [is [some [more tapioca]]]] [some noodles]]]	23 (1.6%)
[Is [[Fraser [who [is [some [more tapioca]]]] [having coffee]]]	22 (1.5%)
[Is [[Fraser [who [is that]]] going]]	20 (1.4%)
[Is [[Fraser [who [is [some [more tapioca]]]] going]]	18 (1.3%)
[Is [[that [who [is that]]] [some noodles]]]	7 (0.50%)
[Is [[that [who [is that]]] going]]	3 (0.21%)

Table 12. Ten most frequently generated AUX-questions of the correct pattern with their bracketings together with their frequencies and their percentage from the total number of sentences of the correct pattern.

Finally, we also investigated the effects of the depth and the absence of discontiguous subtrees on predicting the correct auxiliary fronting by our random generation method. For each maximum subtree depth, we generated 10 million sentences as before by derivations of 3 subtrees, except for maximum subtree depths 1 and 2, for which the shortest derivations that could generate the correct AUX-pattern consisted respectively of 11 and 5 subtrees. For maximum subtree depths 1 and 2, we therefore generated (10 million) sentences by randomly selecting resp. 11 and 5 subtrees, for which at least two subtrees had to contain the word *is* and at least one subtree had to contain the word *who*. For maximum subtree depth 3 and larger, there was always a shortest derivation of 3 subtrees that could generate the correct auxiliary fronting. Next, we checked which was the most frequently generated AUX-pattern for each maximum depth. Table 13 lists for each maximum subtree depth: (1) the length of the shortest derivation, (2) whether or not the correct AUX-pattern was predicted by the MPSD using all subtrees (followed by the predicted pattern), (3) as under (2) but now with only contiguous subtrees.

Maximum Subtree Depth	Length of shortest derivation	Correct AUX-fronting? (all subtrees)	Correct AUX-fronting? (contiguous subtrees only)
1 (PCFG)	11	NO: *Is P Q R who is?	NO: *Is P Q R who is?
2	5	NO: *Is P Q who is R?	NO: *Is P Q who is R?
3	3	NO: *Is P Q who is R?	NO: *Is P Q R who is?
4	3	**YES**: Is P who is Q R?	NO: *Is P Q R who is?
5	3	**YES**: Is P who is Q R?	NO: *Is P Q who is R?
6	3	**YES**: Is P who is Q R?	**YES**: Is P who is Q R?
All (DOP)	3	**YES**: Is P who is Q R?	NO: *Is P Q who is R?

Table 13. Effect of subtree depth and discontiguous subtrees on predicting the correct AUX-fronting. For each maximum subtree depth the table gives: (1) the length of the shortest derivation that can generate the correct AUX-pattern, (2) whether or not the correct AUX-pattern was predicted by the MSPD using all subtrees (together with the predicted pattern), (3) as under (2) using only contiguous subtrees.

The table shows that in order to generate the correct auxiliary fronting we need to include discontiguous subtrees of depth 4, which supports our 'logical' argument in Section 6.1 where also (discontiguous) subtrees of up to depth 4 were needed (Figure 19). Note that if only contiguous subtrees are used in the generation process, the correct AUX-fronting is almost never produced, and the only correct prediction at subtree-depth 6 seems to be anomalous. These results support our previous results on constraining subtree depth and discontiguity in Sections 4 and 5. Although for auxiliary fronting subtrees of maximum depth 4 suffice, we have shown in Section 5.3 that even larger subtrees are needed to predict the correct structures for Eve's longer utterances.

As a matter of precaution, we should keep in mind that Eve does not generate any complex auxiliary fronting construction in the corpus -- but she *could* have done so by combining chunks from her own language experiences using simple substitution. This loosely corresponds to the observation that auxiliary fronting (almost) never occurs in spontaneous child language, but that it can be easily elicited from children (as e.g. in Ambridge et al. 2008).

Auxiliary fronting has been previously dealt with in other probabilistic models of structure learning. Perfors et al. (2006) show that Bayesian model selection can choose the right grammar for auxiliary fronting. Yet, their problem is different in that Perfors et al. start from a set of given grammars from which their selection model has to choose the correct one. Our logical analysis in Section 6.1 is more similar to Clark and Eyraud (2006) who show that by distributional analysis in the vein of Harris (1954) auxiliary fronting can be correctly predicted from the same sentences as used in Section 6.1 (which are in turn taken from MacWhinney 2005). However, Clark and Eyraud do not test their model on a corpus of child language or child-directed speech. More importantly, perhaps, is that Clark and Eyraud show that their model is equivalent to a PCFG, whereas our experiments indicate that subtrees of up to depth 4 are needed to learn the correct auxiliary fronting from the Eve corpus. Of course it may be that auxiliary fronting can be learned by a non-binary PCFG with rich lexical-syntactic categories (which we have not tested in this paper). But it is well-known that PCFGs are inadequate for capturing large productive units and their grammatical structure at the same time. For example, for a PCFG to capture a multi-word unit like *Everything you always wanted to know about X but were afraid to ask*, we need to take this entire expression as right-handside of the PCFG-rule. While such a PCFG can thus recognize this long multi-word unit, it would completely neglect the internal structure of the expression. A PTSG is

more flexible in this respect, in that it allows for productive units that include both the full expressions as well as their syntactic structure. We could enhance PCFGs by cleverly indexing its rules such that the relation between the various rules can be remembered as in a PTSG-subtree. But then we actually obtain a "PCFG"-encoding of a PTSG as explained in the Appendix. (For a mathematical proof that the class of PTSGs is actually stochastically stronger than the class of PCFGs, see Bod 1998: 27ff.)

Auxiliary fronting has also been dealt with in non-hierarchical models of language. For example, Lewis and Elman (2001) and Reali and Christiansen (2005) have shown that auxiliary fronting can be learned by linear processing models. Lewis and Elman trained a simple recurrent network (SRN), while Reali and Christiansen used a trigram model that could predict the correct auxiliary fronting. However, it is not clear what these models learn about the structure-dependent properties of auxiliary fronting since trigram models do not learn structural relations between words. Kam et al. (2005) argue that some of the success of Reali and Christiansen's models depend on 'accidental' English facts. The U-DOP/DOP approach, instead, can learn both the correct auxiliary fronting and its corresponding (unlabeled) syntactic structure. More than that, our method learned the abstract auxiliary fronting rule for complex interrogatives (sentence 18) from the original complex declarative (sentence 16) and a simple interrogative (sentence 17). Simple recurrent networks and trigram models miss dependencies between words when they are separated by arbitrarily long sequences of other words, while such dependencies are straightforwardly captured by PTSGs.

It would be interesting to investigate whether U-DOP/DOP can also simulate auxiliary fronting in other languages, such as Dutch and German that have verb final word order in relative clauses. And there is a further question whether our approach can model children's questions in general, given an appropriate corpus of child utterances (see e.g. Rowland 2007). Research into this direction will be reported in due time.

7 Conclusion

The experiments in this paper should be seen as a first investigation of U-DOP/DOP's simulation of (child) language behavior. As a general model of language learning, our approach is of course too limited and needs to be extended in various ways. The learning of lexical and syntactic categories may be one of the most urgent extensions. Previous work has noted that category induction is a relatively easier task than structure induction (Klein and Manning 2005; Redington et al. 1998). Yet it is not trivial to integrate category learning in the U-DOP model in an incremental way. In principle, the U-DOP approach can be generalized to category learning as follows: assign initially all possible categories to every node in all possible trees (from a finite set of n abstract categories $C_1 \ldots C_n$) and let the MPSD decide which are best trees corresponding to the best category assignments. But apart from the computational complexity of such an approach, it neglects the fact that categories can change quite substantially in the course of child language acquisition. Experiments with incremental category learning will have to await future research.

A major difference between our model and other computational learning models is that we start out with the notion of tree structure, but since we do not know which tree structures are correct, we allow for all of them and let the notion of structural analogy decide. Thus we implicitly assume that the language faculty has prior knowledge about constituent structure, but no more than that. We have seen that our use of tree structures allows for capturing linguistic phenomena that are reliant on non-adjacent, discontiguous dependencies. Other approaches are often limited to contiguous dependencies only, either in learning (Klein and Manning 2005) or in generation (e.g.

Freudenthal et al. 2007). We have not yet evaluated our approach against some other learning models such as Solan et al. (2005) and Dennis (2005) mainly because these models use test corpora different from ours. We hope that our work motivates others to test against the (annotated) Eve corpus as well.

Finally, it may be noteworthy that while U-DOP presents a usage-based approach to language learning, U-DOP's use of recursive trees has a surprising precursor: Hauser, Chomksy and Fitch (2002) claim that the core language faculty comprises just recursion and nothing else. If we take this idea seriously, then U-DOP may be the first computational model that instantiates it. U-DOP's trees encode the ultimate notion of recursion where every label can be recursively substituted for any other label. All else is analogy.

Acknowledgments

Many thanks to Gideon Borensztajn, Remko Scha and Jelle Zuidema for helpful comments on a previous version of this paper. Special thanks go to Stefan Frank whose comments and suggestions were particularly helpful. All remaining errors and unclarities are my responsibility.

Appendix: Computing the MPSD

There is an extensive literature on the computational properties of DOP and U-DOP (see e.g. Sima'an 1996; Scha et al. 1999; Goodman 2003; Bod 2006b, 2007a; Zuidema 2007). This appendix summarizes the main results of U-DOP/DOP's computational background, and focuses on an efficient and compact PCFG reduction of DOP.

The way (U-)DOP combines subtrees into new trees is formally equivalent to a Tree-Substitution Grammar or TSG, and its probabilistic extension is equivalent to a Probabilistic TSG or PTSG (Bod 1998). There are standard algorithms that compute the tree structures (a packed parse forest) of an input string given a PTSG. These algorithms run in Gn^3 time, where G is the size of the grammar (the number of subtrees) and n is the length of the input string (the number of words). Existing parsing algorithms for context-free grammars or CFGs, such as the CKY algorithm (Younger 1967), can be straightforwardly extended to TSGs by converting each subtree t into a context-free rewrite rule where the *root* of t is rewritten by its yield: $root(t) \rightarrow yield(t)$. Indices are used to link each rule to its original subtree. Next, the MPSD can be computed by a best-first beam search technique known as Viterbi optimization (Manning and Schütze 1999). However, the direct application of these techniques to DOP and U-DOP is intractable because the number of subtrees grows exponentially with the number of nodes in the corpus (Sima'an 1996). Goodman (1996, 2003) showed that the unwieldy DOP grammar can be reduced to a compact set of indexed PCFG-rules which is *linear* rather than exponential in the number of nodes in the corpus. Goodman's PCFG reduction was initially developed for the probabilistic version of DOP but it can also be applied to computing the shortest derivation, as we will see below.

Goodman's method starts by assigning every node in every tree a unique number which is called its address. The notation $A@k$ denotes the node at address k where A is the nonterminal labeling that node. A new nonterminal is created for each node in the training data. This nonterminal is called A_k. Let a_j represent the number of subtrees headed by the node $A@j$, and let a represent the number of subtrees headed by nodes with nonterminal A, that is $a = \Sigma_j\, a_j$. Then there is a 'PCFG' with the following property: for every subtree in the training corpus headed by A, the grammar will generate an isomorphic subderivation with probability $1/a$. For example, for a

node ($A@j$ ($B@k$, $C@l$)), the following eight rules are generated, where the number in parentheses following a rule is its probability:

$A_j \rightarrow BC$	$(1/a_j)$	$A \rightarrow BC$	$(1/a)$
$A_j \rightarrow B_kC$	(b_k/a_j)	$A \rightarrow B_kC$	(b_k/a)
$A_j \rightarrow BC_l$	(c_l/a_j)	$A \rightarrow BC_l$	(c_l/a)
$A_j \rightarrow B_kC_l$	(b_kc_l/a_j)	$A \rightarrow B_kC_l$	(b_kc_l/a)

It can be shown by simple induction that this construction produces derivations isomorphic to DOP derivations with equal probability (Goodman 2003: 130-133). It should be kept in mind that the above reduction is not equivalent to a standard PCFG (cf. Manning and Schütze 1999). Different from standard PCFGs, the 'PCFG' above can have several derivations that produce the same tree (up to node relabeling). But as long as no confusion arises, we will refer to this reduction as a 'PCFG-reduction of DOP' and refer to the rules above as 'indexed PCFG rules'. Goodman (2003) also shows that similar reduction methods exist for DOP models in which the number of lexical items or the size of the subtrees are contrained.

Note that the reduction method can also be used for computing the shortest derivation, since the most probable derivation is equal to the shortest derivation if each subtree is given equal probability. This can be seen as follows. Suppose we give each subtree a probability p, e.g. 0.5, then the probability of a derivation involving n subtrees is equal to p^n, and since $0<p<1$ the derivation with the fewest subtrees has the greatest probability.

While Goodman's reduction method was developed for supervised DOP where each training sentence is annotated with exactly one tree, the method can be easily generalized to U-DOP where each sentence is annotated with all possible trees stored in a shared parse forest or packed chart (Billot and Lang 1989). A shared parse forest is usually represented by an AND-OR graph where AND-nodes correspond to the usual parse tree nodes, while OR-nodes correspond to distinct subtrees occurring in the same context. In Bod (2006b, 2007a), Goodman's reduction method is straightforwardly applied to shared parse forests by assigning a unique addresses to each node in the parse forest, just as with the supervised version of DOP.

The shortest derivation(s) and the most probable tree, and hence the MPSD, can be efficiently computed by means of standard best-first parsing algorithms. As explained above, by assigning each subtree equal weigth, the most probable derivation becomes equal to the shortest derivation, which is computed by a Viterbi-based chart parsing algorithm (see Manning and Schütze 1999: 332ff). Next, the most probable tree is equal to the sum of the probabilities of all derivations, which can be estimated by k-best parsing (Huang and Chiang 2005). In this paper, we set the value k to 1,000, which means that we estimate the most probable tree from the 1,000 most probable derivations (in case the shortest derivation is not unique). However, in computing the 1,000 most probable derivations by means of Viterbi it is often prohibitive to keep track of all subderivations at each edge in the chart. We therefore use a simple pruning technique (as in Collins 1999) which deletes any item with a probability less than 10^{-5} times of that of the best item from the chart.

References

Abbot-Smith, K. and Tomasello, M. 2006. Exemplar-learning and schematization in a usage-based account of syntactic acquisition. *The Linguistic Review*, 23, 275-290.

Ambridge, B., C. Rowland and J. Pine, 2008. Is structure dependence an innate constraint? New experimental evidence from chilren's complex-question production. *Cognitive Science* 32, 222-255.
Barlow, M. and S. Kemmer (eds.) 2000. *Usage-Based Models of Language*. CSLI Publications, Stanford.
Bates, E. and J. Goodman, 1999. On the emergence of grammar from the lexicon. In MacWhinney, B. (ed.) 1999. *The Emergence of Language*. Mahwah: Lawrence Erlbaum Associates, 29-79.
Billot, S. and B. Lang, 1989. The Structure of Shared Forests in Ambiguous Parsing. *Proceedings ACL 1989*.
Bod, R. 1992. A Computational Model of Language Performance: Data-Oriented Parsing. *Proceedings COLING 1992*, 855-859.
Bod, R. 1998. *Beyond Grammar: An Experienced-Based Theory of Language*. CSLI Publications, Stanford.
Bod, R. 1999. Context-Sensitive Spoken Dialogue Processing with the DOP Model. *Natural Language Engineering* 5(4), 309-323.
Bod, R. 2000. Parsing with the Shortest Derivation. *Proceedings COLING 2000*, Saarbruecken, 69-75.
Bod, R. 2001. Sentence Memory: Storage vs. Computation of Frequent Sentences. CUNY Conference on Sentence Processing 2001, Philadelphia, PA.
Bod, R. 2002. A Unified Model of Structural Organization in Language and Music. *Journal of Artificial Intelligence Research*, 17, 289-308.
Bod, R. 2003. Do All Fragments Count? *Natural Language Engineering*, 9(4), 307-323.
Bod, R. 2006a. Exemplar-Based Syntax: How to Get Productivity from Examples. *The Linguistic Review* 23, 291-320.
Bod, R. 2006b. An All-Subtrees Approach to Unsupervised Parsing. *Proceedings ACL-COLING 2006*, Sydney, 865-872.
Bod, R. 2007a. Is the End of Supervised Parsing in Sight? *Proceedings ACL 2007*. Prague, 400-407.
Bod, R. 2007b. A Linguistic Investigation into U-DOP. *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*. ACL 2007, 1-8.
Bod, R. and R. Kaplan, 2003. A DOP Model for Lexical-Functional Representations. *Proceedings ACL-COLING 1998*, Montreal.
Bod, R. Scha and K. Sima'an (eds.) 2003. *Data-Oriented Parsing*, The University of Chicago Press.
Bonnema, R., R. Bod and R. Scha, 1997. A DOP Model for Semantic Interpretation. *Proceedings ACL/EACL 1997*, Madrid, Spain, 159-167.
Borensztajn, G., J. Zuidema and R. Bod, 2008. Children's grammars grow more abstract with age – Evidence from an automatic procedure for identifying the productive units of language. To appear in *Proceedings CogSci 2008*, Washington D.C. (accepted for publication).
Brown, R. 1973. *A First Language: The Early Stages*. George Allen & Unwin Ltd., London.
Bybee, J. 2006. From Usage to Grammar: The Mind's Response to Repetition. *Language* 82(4), 711-733.
Bybee, J. and P. Hopper 2001. *Frequency and the Emergence of Linguistic Structure*. John Benjamins.
Carroll, J. and D. Weir, 2000. Encoding Frequency Information in Stochastic Parsing Models. In H. Bunt and A. Nijholt (eds.), *Advances in Probabilistic Parsing and Other Parsing Technologies*, 13-28.
Chater, N. 1999. The Search for Simplicity: A Fundamental Cognitive Principle? *The Quarterly Journal of Experimental Psychology*, 52A(2), 273-302.
Chi, Z. and S. Geman 1998. Estimation of probabilistic context-free grammars. *Computational Linguistics* 24(2), 299-305.
Chiang, D. 2007. Hierarchical phrase-based translation. *Computational Linguistics* 33(2):201–228
Chomsky, N. 1965. *Aspects of the Theory of Syntax*. The MIT Press.
Chomsky, N. 1966. *Cartesian Linguistics.* New York, Harper & Row.
Chomsky, N. 1971. *Problems of Knowledge and Freedom*. Pantheon Books.
Clark, A. 2000. Inducing syntactic categories by context distribution clustering. *Proceedings CONLL 2000*, 91-94.
Clark, A. 2001. Unsupervised induction of stochastic context-free grammars using distributional clustering. *Proceedings CONLL 2001*, 105-112.
Clark, A. and R. Eyraud, 2006. Learning Auxiliary Fronting with Grammatical Inference. *Proceedings CONLL 2006*, New York.
Clark, B. 2005. On Stochastic Grammar. *Language* 81, 207-217.
Collins, M. 1999. *Head-Driven Statistical Models for Natural Language Processing*. PhD thesis. University of Pennsylvania.
Collins M. and N. Duffy 2002. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. *Proceedings ACL 2002*, Philadelphia, PA.
Conway, C. and N. Christiansen, 2006. Statistical Learning within and between Modalities. *Psychological Science* 17(10), 905-914.

Crain, S. 1991. Language Acquisition in the Absence of Experience. *Behavorial and Brain Sciences* 14, 597-612.
Crain, S. and M. Nakayama, 1987. Structure Dependence in Grammar Formation. *Language* 63, 522-543.
Crain, S. and R. Thornton, 2007. Acquisition of Syntax and Semantics. In M. Traxler and M. Gernsbacher (eds.), *Handbook of Psycholinguistics*, Elsevier. In press.
Croft, B. 2001. *Radical Construction Grammar*. Oxford University Press.
Dennis, S. 2005. An exemplar-based approach to unsupervised parsing. *Proceedings of the Twenty Seventh Conference of the Cognitive Science Society*.
Esper, E. 1973. *Analogy and Association in Linguistics and Psychology*. University of Georgia Press.
Frazier, L. 1978. *On Comprehending Sentences: Syntactic Parsing Strategies*. PhD. Thesis, University of Connecticut.
Freudenthal, D., J. Pine, J. Aguado-Orea and F. Gobet 2007. Modelling the developmental patterning of finiteness marking in English, Dutch, German and Spanish using MOSAIC. *Cognitive Science*, 31, 311-341.
Genter, D. and A. Markman, 1997. Structure Mapping in Analogy and Similarity. *American Psychologist* 52(1), 45-56.
Goldberg, A. 2006. *Constructions at Work: the nature of generalization in language*. Oxford University Press.
Goodman, J. 1996. Efficient algorithms for parsing the DOP model. *Proceedings Empirical Methods in Natural Language Processing 1996*, Philadelphia, PA: 143-152.
Goodman, J. 2003. Efficient parsing of DOP with PCFG-reductions. In R. Bod, R. Scha and K. Sima'an (eds.). *Data-Oriented Parsing*, CSLI Publications, 125-146.
Harris, Z. 1954. Distributional Structure. *Word* 10, 146-162.
Hauser, M., N. Chomsky and T. Fitch, 2002. The Faculty of Language: What Is It, Who Has It, and How Did It Evolve?, *Science* 298, 1569-1579.
Hearne, M. and A. Way 2004. Data-oriented parsing and the Penn Chinese Treebank. *Proceedings 1st Intl. Joint Conf. Natural Language Processing*, May, Hainan Island, 406-413.
Hoogweg, L. 2003. Extending DOP with Insertion. In R. Bod, R. Scha and K. Sima'an (eds.). *Data-Oriented Parsing*, CSLI Publications, 317-335.
Huang, L. and D. Chiang 2005. Better *k*-best parsing. *Proceedings IWPT 2005*, 53-64.
Johnson, M. 2002. The DOP estimation method is biased and inconsistent. *Computational Linguistics*, 28(1): 71-76.
Joshi, A. 2004. Starting with complex primitives pays off: complicate locally, simplify globally. *Cognitive Science* 28, 637-668.
Jurafsky, D. 2003. Probabilistic Modeling in Psycholinguistics. In Bod, R., J. Hay and S. Jannedy (eds.), *Probabilistic Linguistics*, The MIT Press, 39-96.
Kam, X., I. Stoyneshka, L. Tornyova, W. Sakas and J. Fodor 2005. Statistics vs. UG in Language Acquisition: Does a Bigram Analysis Predict Auxiliary Inversion? *Proceedings 2nd Workshop on Psychocomputational Models of Human Language Acquisition*. ACL, 69-71.
Kaplan, R. 1996. A Probabilistic Approach to Lexical-Functional Analysis. *Proceedings of the 1996 LFG Conference and Workshops*. Stanford: CSLI Publications.
Kay, M. 1980. *Algorithmic Schemata and Data Structures in Syntactic Processing*. Report CSL-80-12, Xerox PARC, Palo Alto, Ca.
Kay, P. and C. Fillmore 1999. Grammatical constructions and linguistic generalizations: the What's X doing Y? construction. *Language*, 75, 1-33.
Klein, D. and C. Manning 2002. A general constituent-context model for improved grammar induction. *Proceedings ACL 2002*, Philadelphia.
Klein, D. and C. Manning 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. *Proceedings ACL 2004*, Barcelona.
Klein, D. and C. Manning 2005. Natural language grammar induction with a generative constituent-context model. *Pattern Recognition* 38, 1407-1419.
Langacker, R. 1987. *Foundations of cognitive grammar: Theoretical Prerequisites*. Stanford, CA: Stanford University Press.
Lewis, J. and J. Elman 2001. Learnability and the statistical nature of language: Poverty of stimulus arguments revisited. *Proceedings of 26th annual Boston Univ. Conference on Language Development*, 359-370.
MacWhinney, B. 1978. The acquisition of morphophonology. *Monographs of the Society for Research in Child Development 43.*
MacWhinney, B. 2000. *The CHILDES Project: Tools for Analyzing Talk.* Lawrence Erlbaum Associates, Mawah, NJ.
MacWhinney, B. 2005. Item-based Constructions and the Logical Problem. *Proceedings of the Second Workshop on Psychocomputational Models of Human Language Acquisition*, Ann Arbor.
Manning, C. and H. Schütze 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Marcus, M., B. Santorini and M. Marcinkiewicz, 1993. Building a Large Annotated Corpus of English: the Penn Treebank, *Computational Linguistics* 19(2), 302-330.

Moerk, E., 1983. *The Mother of Eve as a First Language Teacher*. ABLEX, Norwood, N.J.

Neumann, G. and D. Flickinger 2002. HPSG-DOP: data-oriented parsing with HPSG. *Proceedings 9th International Conference on HPSG* (HPSG-2002), Seoul.

Perfors, A., Tenenbaum, J., Regier, T. 2006. Poverty of the Stimulus? A rational approach. *Proceedings 28th Annual Conference of the Cognitive Science Society*. Vancouver.

Peters, A. 1983. *The units of language acquisition*. Cambridge University Press.

Pinker, S. 1999. *Words and Rules: The Ingredients of Language*. London: Widenfeld and Nicolson.

Pullum, G. and B. Scholz 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review* 19, 9-50.

Reali, F. and M. Christiansen 2005. Uncovering the richness of the stimulus: structure dependence and indirect statistical evidence. *Cognitive Science* 29, 1007-1028.

Redington, M., Chater, N. and Finch, S. 1998. Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive Science* 22, 425-469.

Rowland, C. 2007. Explaining errors in children's questions. *Cognition* 104, 106-134.

Sagae, K., E. Davis, A. Lavie, B. MacWhinney and Shuly Wintner 2007. High-accuracy Annotation and Parsing of CHILDES Transcript. *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*. ACL 2007.

Sawyer, J. 2001. Bifurcating the verb particle construction. Evidence from child language. *Annual Review of Language Acquisition* 1, 119-156.

Scha, R. 1990. Taaltheorie en Taaltechnologie; Competence en Performance, in Q. de Kort and G. Leerdam (eds), *Computertoepassingen in de Neerlandistiek*, Almere: Landelijke Vereniging van Neerlandici (LVVN-jaarboek).

Scha, R., Bod, R. and Sima'an, K. 1999. A memory-based model of syntactic analysis: Data-oriented parsing. *Journal of Experimental & Theoretical Artificial Intelligence*, 11, 409-440.

Sima'an, K. 1996. Computational complexity of probabilistic disambiguation by means of tree grammars. *Proceedings COLING 1996*, 1175-1180.

Skousen, R. 1989. *Analogical Modeling of Language*. Dordrecht: Kluwer.

Skut, W., B. Krenn, T. Brants and H. Uszkoreit 1997. An annotation scheme for free word order languages. *Proceedings ANLP 1997.*

Solan, D., D. Horn, E. Ruppin, and S. Edelman 2005. Unsupervised learning of natural languages. *Proceedings National Academy of. Science*, 102:11629-11634.

Tomasello, M. 2003. *Constructing a Language*. Harvard University Press.

Xia, F. and M. Palmer 2001. Converting Dependency Structures To Phrase Structures. *Proceedings HLT 2001*, San Diego.

Xue, N., F. Chiou and M. Palmer 2002. Building a large-scale annotated Chinese corpus. *Proceedings COLING 2002*, Taipei.

Younger, D. 1967. Recognition and parsing of context-free languages in time n3. *Information and Control*, 10(2), 189-208.

van Zaanen, M. 2000. ABL: Alignment-Based Learning. *Proceedings COLING 2000*, Saarbrücken.

Zollmann, A. and K. Sima'an 2005. A consistent and efficient estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics*, Vol. 10 (2005) Number 2/3, 367-388.

Zuidema, W. 2006. Theoretical Evaluation of Estimation Methods for Data-Oriented Parsing. *Proceedings EACL 2006*, Trento, Italy.

Zuidema, W. 2007. Parsimonious data-oriented parsing. *Proceedings EMNLP 2007*, 551-560.

Modeling language acquisition, change and variation

Willem Zuidema
Language Evolution and Computation Research Unit
School of Philosophy, Psychology and Language Sciences
and Institute of Animal, Cell and Population Biology
University of Edinburgh
40, George Square
Edinburgh EH8 9LL, United Kingdom
jelle@ling.ed.ac.uk

http://www.ling.ac.uk/ $\sim$ jelle

Abstract

The relation between Language Acquisition, Language Change and Language Typology is a fascinating topic, but also one that is difficult to model. I focus in this paper on the question how theories of language acquisition constrain theories of language change and typology. In the generative tradition and "Learnability Theory" this problem is approached by assuming that all linguistic variation can be described in terms of a relatively small number of parameters of a universal, innate core, the Universal Grammar. In this view, language acquisition is parameter setting, and language change is parameter change. I review some simple acquisition models and their consequences for language change, and discuss some problems with this approach. I will then discuss an alternative approach that is based on "Explicit Induction" algorithms for grammatical formalisms. I discuss which approach is most useful for which problems.

1 Language acquisition, change and typology

Every healthy human infant is capable of acquiring any one of a dazzling variety of human languages. This simple fact poses two fundamental challenges for linguistics: (1) understanding how children are so extremely successful at this apparently complex task, and (2) understanding how, although all humans have such similar linguistic abilities, such a wide variety of languages has emerged. These challenges are intricately linked: the languages that we observe today, are the result of thousands of years of cultural transmission, where every generation has acquired its language from the observed use by previous generations. That makes the acquisition of language a rather unique learning problem for learning theory, because what is being learned is itself the result of a learning process. Conversely, the structure of a language (say modern English) at any one time (say, 2003) is the result of perhaps millions of individuals learning from examples from a language with a very similar structure (say, the English of the 1960s).

This so-called *circular causality* (Steels, 1999) makes the relation between Language Acquisition, Language Change and Language Typology a fascinating topic, but also one that is difficult to model. I will focus here on the question how theories of language acquisition constrain theories of language change and typology. In the generative tradition and "Learnability Theory" this problem is approached by assuming that all linguistic variation can be described in terms of a relatively small number of parameters of a universal, innate core, the Universal Grammar. In this view, language acquisition is parameter setting, and

language change is parameter change. In the following I will review some simple acquisition models and their consequences for language change, and discuss some problems with this approach. I will then discuss an alternative approach from the emerging field of computational modeling of the evolution of language (Kirby, 2002b), that is based on "Explicit Induction" algorithms for grammatical formalisms. I will argue that the differences between the two approaches have been exaggerated, and will discuss for which sort of problems which sort of approach is most useful.

2 Parameter models

The "Parameter change" approach to this problem is based on *parameterizing* linguistic structure, such that we can characterize all differences between possible human languages by a vector of a small number of parameters. E.g., in the Principles and Parameters approach (Chomsky, 1981; Bertolo, 2001), language acquisition is described in terms of parameter settings for a universal core, the Universal Grammar. With such a description of language in hand, we can reformulate the challenges as follows: (1) how can learning, given primary linguistic data that conforms to any particular set of parameters, find that set of parameters? (2) given a set of learning procedures that are capable of finding the correct parameters, which ones predict the type of language change and statistical distributions (universals tendencies, Kirby 1999) that we can actually observe?

2.1 Parameter setting

In the "parameter setting" models of language acquisition, one assumes a finite number N of possible grammars. If all variation can be described by n different, Boolean and independent parameters, such that the total number of possible grammars is $N = 2^n$. Such parameters determine, for instance, whether or not an object precedes the main verb in a sentence, or whether or not the subject can be left out. Typically, although the number of parameters is estimated at around 30, concrete examples are only worked out for the 2 or 3 least controversial proposed parameters. A lot of work in parameter setting works with rather simplified models that can be studied analytically, and that depend only on the finiteness of N. Examples of such models are "memory-less learning", "batch learning" (e.g. Nowak *et al.*, 2001) and "learning by enumeration" (Gold, 1967). It is useful to look in a bit more detail at these models.

Memory-less learning (Niyogi, 1998) is arguably the simplest language acquisition model. The algorithm works by choosing a random grammar from the set of possible grammars each time the input data shows that the present hypothesis is wrong. The algorithm obviously is not very efficient, because it can arrive at hypotheses it has already rejected before; i.e. each time it randomly chooses a new grammar, it forgets what it has learned from all data it has received before. This algorithm is only of interest because it is simple and provides a lower bound on the performance of any reasonable algorithm (Nowak *et al.*, 2001).

The *batch learner*, in contrast, memorizes all received sentences and finds all grammars from the set of possible ones that are consistent with these sentences. Equivalently, it keeps track of all possible grammars that are still consistent with the received data. In any case, for any reasonably large set of possible grammars, the batch learner has monstrous memory and processing requirements. Its value lies in the fact that it is simple, and provides an upper bound on the performance of any reasonable learning algorithm, as long as there is no a-priori reason to prefer one grammar that is consistent with the data over another.

As exemplified by appendix A, we can, with a bit of effort, derive explicit formulas that describe the probability of success q as a function of the number of input sentences for both the memory-less and the batch learner. Under the assumption that every wrong grammar is equally similar to the right grammar (described with a similarity parameter a), we can in fact give a complete transition matrix T, where all

diagonal values are $q_{memoryless}$ and all off-diagonal values are $(1 - q_{memoryless})/(N - 1)$. This transition matrix plays an important role in models of language change described in the next section.

It is important to realize that these algorithms only work because a finite (and in fact, relatively small) number of possible grammars is assumed. Moreover, calculations such as in appendix A are relatively easy due to some important assumptions: (1) that the algorithms are not biased at all to favor certain possible grammars over others; (2) that (in the case of the memory-less learner) the probability of jumping to a wrong or right grammar remains constant throughout the learning process; and (3) that all grammars are equally similar to each other. Without these assumptions, similar calculations quickly get rather complex.

For instance, *learning by enumeration* (Gold, 1967), as the name suggests, proceeds by enumerating one at a time, and in prespecified order all possible grammars. Only if a grammar is inconsistent with incoming data ("text"), does the algorithm move on to the next grammar. The procedure is of interest, because it can be used as a criterion for learnability (Gold, 1967)[1]. Calculating q is more difficult than before, because the probably of changing to a wrong grammar *decreases* over time.

The *trigger learning algorithm* (Wexler & Culicover, 1980) is a popular model that is of (slightly) more practical interest. Rather than picking a random new grammar, as the memory-less learner does, or enumerating grammars in a random order, as in learning by enumeration, it changes a random parameter when it finds an input sentence that is inconsistent with the present hypothesis. If with the new parameter setting the sentence can be parsed, the change is kept, otherwise it is reverted. The trigger learning algorithm thus implements a kind of hill-climbing (gradient ascent), by keeping parameters that do well and only making a small change when it improves performance. The probability of the trigger algorithm to give the right grammar after b sentences is even more tricky to calculate, because the probability to reject a wrong hypotheses *decreases* as more and more parameters get correctly set.

Many other parameter setting models exists. E.g. Briscoe (2002a) develops a variant of the trigger learning algorithm, where parameters are no longer independent, but fall into linguistically motivated inheritance hierarchies. Further, rather than choosing a single parameter at random and changing it, as in the TLA, Briscoe's algorithm selects several random parameters and keeps track of their most likely setting in a Bayesian, statistical fashion. Yang (2000) argues that language acquisition is best viewed as a selectionist process, where many different parameter sets are considered in parallel. Niyogi & Berwick (1995) and Yang (2000) consider the further complication that children learn from input sentences that are drawn from different languages, and explore the expectations on what grammar settings they will end up with. In all these models, calculating the probabilities of the outcome of learning gets very complex and results are typically obtained by using computer simulations.

2.2 Parameter change

Niyogi & Berwick (1995), as well as neural network modelers Hare & Elman (1995), argue that a theory of language acquisition – and the mistakes children make when confronted with insufficient or ambiguous input – implies a theory of language change. Similarly, Kirby (1999) explores the idea that a theory on language use and processing – which alter the primary linguistic data – leads to specific expectations on language change and the resulting linguistic variation. Hence, by working out the consequences for language change and comparing them to empirical data, theories on language use, processing and acquisition can be

[1]Learning by enumeration can, within finite time, find the target grammar from a class of grammars if the following conditions hold: (1) the class of grammars is finite (enumerable), (2) for every two grammars in the class, there exists a sentence that distinguishes between two grammars (i.e. that is grammatical according to one, and ungrammatical according to the other), and (3) the distinguishing sentence will occur within a finite amount of time in the the text, generated by the target grammar. It follows that the class of grammars is then learnable from text. It can be shown that superfinite classes of grammars, such as the context-free or context-sensitive grammars, are not learnable in this sense (Gold, 1967). Principles & Parameters-models, in contrast, are learnable (Wexler & Culicover, 1980) and so are many other classes (Angluin, 1980).

tested. Formally, a class of grammars $\mathcal{G}$, a learning algorithm $\mathcal{A}$ and a model of the primary linguistic data (a probability distribution $\mathcal{P}_i$ over the possible sentences of language i) together constitute the main ingredients of a dynamical system that describes the change in numbers of speakers of each language[2].

Several general results have been obtained. For instance, Niyogi & Berwick (1995) and Yang (2000) find that with different choices for $\{\mathcal{G}, \mathcal{A}, \mathcal{P}\}$, the change in the number of speakers of a particular language tends to follow an S-shaped curve, consistent with observed patterns in historical data. More interestingly, Nowak *et al*. (2001) derive a *coherence threshold*. In their model, natural selection selecting for more frequent grammars, helps a population to converge on a specific grammar. Mistakes in learning, on the other hand, lead to divergence, because it essentially randomizes the choice of grammars. Nowak et al. find that if the accuracy in learning is below a precise threshold, all coherence in the population is lost and all languages are spoken with equal probability[3].

Niyogi and Berwick apply their methodology to a number of case studies. For instance, they look at a simple 3-parameter system where the parameters determine whether or not specifiers (1) and complements (2) come before the head of a phrase, and whether or not the verb is obligatorily in second position (3). In this system, there are 8 different possible grammars (languages). By making assumptions on the frequency with which triggers for each of the parameters are available to the child, they can estimate the probability a specific learning algorithm can learn each language. They numerically determine the probabilities of transitions between each of the 8 language over 30 generations with 128 triggers per generation. They find that languages with the third parameter set to "0" ($V2-$) are extremely unstable and that the $V2+$ parameter therefore quickly gets fixed in all simulations. This observation is contrary to observed trends in historical data, where $V2+$ is typically lost. Niyogi and Berwick argue that this falsifies their preliminary model, and thus illustrates the feasibility of testing the diachronic accuracy of the assumptions on $\{\mathcal{G}, \mathcal{A}, \mathcal{P}\}$.

2.3 Some features of parameter change models

Several other parameter change models have been studied. They have in common the emphasis on the uniformity of languages, i.e. all possible languages (grammars) are of equal quality. Hence, children acquiring a language do not go from a simple grammar to a more complex one, but rather jump from one grammar to an equally complex alternative. Not the quality of the language, but the uncertainty about which is the correct one changes over time.

Moreover, in all these models the acquisition of syntax is studied independently from the acquisition of phonology, semantics, pragmatics and the lexicon, and, usually, independent from the particularities of the child's parsing algorithm. The training data are "triggers", i.e. strings of grammatical categories. The problems of learning the syntactic categories of words and their meaning, and learning to recognize the phonological form and the boundaries between words are all ignored.

Further, the models fit into a tradition that is much mathematically oriented. Although many results are obtained through numerical simulations, the models are formulated at a rather abstract level. Generations are typically discrete, the number of parameters small (2, 3, 5), number of training samples and the number of individuals in a population very small or, alternatively, infinite.

The models are valuable, because they give a *general* insight in how linguistic conventions can change and spread in a population. However, the problem with this approach is that its potential for explaining *specific* aspects of language acquisition and language typology depends completely on the successful parametrization of linguistic descriptions. That dependence has advantages, because it makes the relation with other linguistic theories very clear, but it has some major disadvantages as well.

[2]In addition to the triple $\{\mathcal{G}, \mathcal{A}, \mathcal{P}\}$ (Niyogi & Berwick, 1995), one needs assumptions on population and generation structure and the number of training sentences the algorithm receives.

[3]Presumably, a similar mechanism explains the lack of coherence in the simulations of Niyogi & Berwick (1995).

First, there is, as for now, no such parametrization available. If efficient parametrization (i.e. with 20 or 30 parameters) turns out to be impossible, models that depend on them will be inadequate. Second, even if it is possible in principle, without a complete theory available on what each parameter means, solutions in terms of these parameters give little insight on why children learn certain things with more ease than others, or why languages tend to show certain patterns more often than others. Finally, parameter-models might give an adequate description of the variation in languages in a quasi-stable state, but that does not necessarily mean that they also give an adequate description of language variety when languages are changing. In particular, observed trends in language change regarding the interaction between phonology, syntax, semantics and pragmatics seem hard to capture in available parameter models.

3 Explicit Induction

3.1 Grammar Induction: impossible and irrelevant?

Grammar Induction algorithms are usually based on the intuition that the frequency of occurrence of substring in the training sentences, and the contexts in which they appear, contain information on what the underlying constituents and the rules of combination of the target grammar are. E.g. Zellig Harris, in describing the methods linguists use to infer the grammar of an unknown language, defines the crucial concept of "substitutability" as follows: "If our informant accepts DA'F as a repetition of DEF, and if we are similarly able to obtain E'BC as equivalent to ABC, then we say that A and E are mutually substitutable" (Zellig Harris, 1951, quoted in van Zaanen 2001).

It is possible to design induction algorithms that, just like Harris's linguist, use observed patterns in training sentences to induce the underlying grammar. However, due to initial negative results on the theoretical possibility of learning a grammar from positive data (Gold, 1967) and developments in linguistic theory (e.g. Chomsky, 1965), the *induction* of grammar has been widely viewed as both impossible and irrelevant.

The supposed impossibility of grammar induction is based on a widespread misinterpretation of negative learnability results. Gold (1967) showed that e.g. the class of context-sensitive languages is not *identifiable in the limit*. Even we if accept identification in the limit as the appropriate criterion for learnability, Gold's results mean nothing more than, in his own words:

> *"The class of possible natural languages is much smaller than one would expect from our present models of syntax. That is, even if English is context-sensitive, it is not true that any context-sensitive language can occur naturally. Equivalently, we may say that the child starts out with more information than that the language it will be presented is context-sensitive. In particular, the results on learnability from text imply the following: The class of possible natural languages if it contains languages of infinite cardinality, cannot contain all languages of finite cardinality." (Gold, 1967)*

In other words, a class of context-sensitive grammars needs to be constrained to make it learnable. Angluin (1980) has shown that very non-trivial classes of formal languages are learnable. Nothing in the formal results, however, proves that the necessary restrictions are due to an extensive, innate, language-specific Universal Grammar; they could be simply generic properties of the human brain[4].

The supposed irrelevance of grammar induction algorithms is based on the fact that the dominant linguistic theories of the last decades assume extensive innate knowledge. If children don't do grammar induction, why design computer programs that do? Evidence for this view comes – in addition to the learnability

[4]Although it is of course true that learnability is a valid test for judging the validity of a (grammatical) theory, and that few proposed theories other than those from the nativist tradition pass it. However, one can argue that nativist theories, rather than solving the learnability problem, simply shift it to the domains of evolutionary theory and cognitive neuroscience.

results – from empirical observations in child language acquisition. Typically, such arguments have the form: the child correctly uses construction X very early in life, even though the primary linguistic data it has received up to that point does not provide enough evidence to choose between X and several alternative logical possibilities. Thus, it is concluded, the child must have prior (innate) knowledge of X.

More and more it is now recognized that this "knowledge of X" might be an emergent result of the interaction between not necessarily language-specific cognitive and learning abilities, and the structure, meaning and pragmatics of the linguistic data the child received (MacWhinney, 1999). Consequently, the need to postulate language-specific adaptations might be limited (Jackendoff, 2002; Hauser *et al.*, 2002).

3.2 Induction Algorithms

Wolff (1982), and similarly Stolcke (1994), Langley & Stromsten (2000) and Zuidema (2003), presents a model based on the idea that a grammar is a compressed representation of a possibly infinite language (string set). These algorithms all use context-free grammars as the grammar formalism, learn from text and run through three phases that can be termed "incorporation", "compression" and "generalization". I will refer to these algorithms as "compression-based induction".

In the incorporation phase, input sentences s are stored as idiosyncratic rewrite rules $S \mapsto s$. In the compression phase (or "syntagmatic merging"), the most frequent substrings z in the right-hand sides of the stored rules are replaced by a unique non-terminal symbol N. Rules of the form $N \mapsto z$ are added to the grammar. In the generalization phase (or "paradigmatic merging"), two nonterminals N and N' are considered *substitutable* if they occur in the same context; all occurrences of N' are then replaced by N. Different variants of the basic algorithm differ in how *greedy* they are, and in whether or not they are *incremental*. Kirby (2000), and later papers, uses a algorithm were the context-free grammars are enriched with a predicate-logic based semantics.

A related framework based on substitutability is developed by van Zaanen (2001) and termed "Alignment Based Learning" (ABL). Van Zaanen develops a number of algorithms for the two phases of the ABL framework: Alignment learning and selection learning. In the alignment learning phase input sentences are compared, aligned and common substrings are identified. The *unequal* parts z and z' of the two sentences are labeled with a non-terminal. The non-terminal is unique if neither z nor z' was labeled already, but the algorithm reuses the existing label if available, and equates the two non-terminals if both z and z' were labeled already. In the latter two conditions a form of generalization occurs. Each labeling is a hypothesis on a possible constituent of the target language, and very many such hypotheses are generated.

In the selection learning phase, a subset of the generated hypotheses is selected. That subset is chosen such that it is concise (each hypothesis can be used to analyze many sentences), and that it is internally consistent (hypotheses do not overlap). The ABL algorithm yields a tree-bank: an annotated version of the input corpus (it thus implements automated tagging). From the tree-bank, context-free grammars can be trivially induced.

3.3 Language Evolution

In the "Explicit Induction" approach to modeling language change and evolution, language change is studied based on similar induction algorithms, i.e. learning algorithms that produce an explicit grammar based on training sentences (see Hurford, 2002, for a review). Such an approach avoids the problems of parameter models, because they can incorporate any available linguistic formalism. However, they have two major disadvantages as well: (1) language induction is very challenging problem that is far from solved, even for simplified and well understood grammar formalisms; (2) models that incorporate a full-blown linguistic formalism, including procedures for language production and interpretation, quickly get very complex.

Two recent models by Kirby (2002a) and Batali (2002) show that there is reason for optimism for progress on bl problems. Kirby presents a model that is very clear in its set-up. It uses first-order predicate logic with a small set of entities and predicates to represent semantics, and a extension of context-free grammars to represent syntax and the syntax-semantics mapping. The model thus uses well-understood and conventional linguistic formalisms and a simple learning procedure. However, by using the output of one learning cycle as input for the next Kirby was able to get some unconventional results: the spontaneous emergence of a recursive, infinite but learnable language. However, the learning algorithm used is very brittle, and it's difficult to extend the model to domains with more diverse semantics and a more heterogeneous syntax.

In contrast, Batali's model is very difficult to understand. It also uses a form of predicate logic to represent semantics, but it uses "exemplars" as the basic representation of the grammar, and "argument maps" to guide the combination of exemplars into meaningful sentences. The results show the emergence of a complex language, with properties similar to case marking and subordinate clause marking in natural languages. The emergent languages are essentially infinite but nevertheless learnable (from meaning–form pairs). The learning algorithm is successful and robust in this complex domain presumably because of the redundancy it allows.

3.4 Some features of explicit induction models

Several other explicit induction models have been studied. They have in common that no uniformity of languages are assumed. Typically, individuals in these models start with an empty grammar and empty lexicon, and gradually add new rules and lexical items based on the received sentences and observed patterns. Individuals are, however, equipped with an invention procedure, such that they can generate new sentences when required.

Further, in these models learning is typically from form–meaning pairs and a lexicon is built-up in parallel with the grammar. The recognition of phonemes and the pragmatics of dialogs are built-in as assumptions of the models.

The models are all implemented as computer programs. Typically, the models are rather concrete: they consist of a population of individuals, with procedures for production, invention, interpretation and induction, and a set of possible message to communicate. The languages studied in these models are still relatively simple, and exhibit just some basic word orders or morphological markers for the semantic roles of agents, patients and action. Empirical data from historical linguistics has so far played no role in these studies.

4 Discussion

I have reviewed some models of language acquisition and language change from two different traditions. The crucial question – which approach is best? – is still largely open to discussion. The following issues are important in comparing both approaches:

Learnability - Theoretical arguments. From the field of learnability theory it has sometimes been argued that grammar induction is impossible. In section 3.1 I have argued that this position is based on a misunderstanding of the negative learnability results. Learnability, however, is an important test for the validity of a grammar formalisms and induction algorithms. The challenge is to find a combination of a formalism that is as expressive as human languages are (i.e. mildly context-sensitive), and a learning algorithm that can induce it from the available primary linguistic data. In my view, parameter setting

models meet this challenge, but only by making unsatisfactory assumptions on the prior knowledge the algorithms start with. Explicit induction models, on the other hand, present considerable progress (i.e. most work with context-free grammars), but more work still needs to be done.

Learnability - Empirical arguments. From the field of psycholinguistics it has been argued that children have prior knowledge of syntactic constructions, because they choose, from apparently many logical possibilities that are consistent with the received evidence, the correct, seemingly arbitrary option. Grammar induction models, in this view, are – if not impossible – irrelevant, because children do not do induction. I believe that explicit induction algorithms have already shown that the logic of this argument is false. There is no need for assuming explicit prior knowledge, because the outcome of the interaction between learning biases and training data is subtle and often unexpected. Moreover, because languages are transmitted culturally from generation to generation, seeming arbitrary choices are likely to be the correct ones, because previous generations have used the same arbitrary learning algorithm to learn their language (Deacon, 1997; Kirby, 2000; Briscoe, 2002a; Zuidema, 2003).

Equivalence More subtly, it has been suggested that explicit induction models might in some sense be equivalent to parameter setting models. If the space of grammars that induction algorithms explore is finite, then that space could in principle be parametrized and hence described by a finite number of parameters. The induction algorithm can then be described, albeit possibly in a clumsy and complicated way, as a parameter setting procedure. If this is true – and it presumably is for the context-free grammar and finite-state machine inducers – the crucial issue is parsimony and clarity. Presumably, for some purposes the representation in terms of parameters is more useful, but for comparison with psycholinguistic, neurological and historical data the explicit grammar representation seems more appropriate. Further, the parameterized representation leads naturally to the uniformity assumptions, whereas the explicit grammar representation leads naturally to the view that grammars grows over time. Finally, stochastic grammar formalisms can not be parametrized in the concise way that parameter setting models usually assume. Worse, lexicalized, exemplar-based models can not be parametrized because there are infinitely many probability distributions that can be assigned to the string set (Bod, 1998).

In conclusion, the two approaches to modeling of language change are rooted in different theoretical positions on the nature of language and language acquisition. If one adopts the Principles and Parameters framework, the parameter change approach is the appropriate way to conceptualize language change. However, this approach requires more work to make explicit how each parameter is to be interpreted, which triggers for each parameter are available, how the child learns her lexicon and recognizes syntactic categories in the sentences it receives, how parameters depend on each other, etc. Moreover, it requires a satisfactory explanation for the evolution and development of the Universal Grammar in the child's brain. However, some Explicit Induction models might, even if one adopts this approach, still be useful as an equivalent representations that can be more easily compared to empirical data.

If one rejects the Uniformity Hypothesis and conceptualizes grammar acquisition as the gradual built-up of a grammar in the mind of the child, explicit induction models are the appropriate approach. Parameter change models are still useful as simple, but mathematically sophisticated models of how conventions spread in a population.

References

ANGLUIN, D. (1980). Inductive inference of formal languages from positive data. *Information and Control* **21**, 46–62.

BATALI, J. (2002). The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In: Briscoe (2002b).

BERTOLO, S., ed. (2001). *Language Acquisition and Learnability*. Cambridge University Press.

BOD, R. (1998). *Beyond Grammar: An experience-based theory of language*. Stanford, CA: CSLI.

BRISCOE, T. (2002a). Grammatical acquisition and linguistic selection. In: Briscoe (2002b).

BRISCOE, T., ed. (2002b). *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press.

CHOMSKY, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.

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

DEACON, T. (1997). *Symbolic species, the co-evolution of language and the human brain*. The Penguin Press.

GOLD, E. M. (1967). Language identification in the limit. *Information and Control (now Information and Computation)* **10**, 447–474.

HARE, M. & ELMAN, J. (1995). Learning and morphological change. *Cognition* **56**, 61–98.

HAUSER, M., CHOMSKY, N. & FITCH, W. (2002). The faculty of language: what is it, who has it, and how did it evolve? *Science* **298**, 1569–1579.

HURFORD, J. R. (2002). Expression / induction models of language. In: Briscoe (2002b).

JACKENDOFF, R. (2002). *Foundations of Language*. Oxford, UK: Oxford University Press.

KIRBY, S. (1999). *Function, selection and innateness: The emergence of language universals*. Oxford University Press.

KIRBY, S. (2000). Syntax without natural selection: How compositionality emerges from vocabulary in a population of learners. In: *The Evolutionary Emergence of Language: Social function and the origins of linguistic form* (Knight, C., Hurford, J. & Studdert-Kennedy, M., eds.). Cambridge, UK: Cambridge University Press.

KIRBY, S. (2002a). Learning, bottlenecks and the evolution of recursive syntax. In: Briscoe (2002b).

KIRBY, S. (2002b). Natural language from artificial life. *Artificial Life* **8**, 185–215.

KOMAROVA, N., NIYOGI, P. & NOWAK, M. (2001). The evolutionary dynamics of grammar acquisition. *J. Theor. Biology* **209**, 43–59.

LANGLEY, P. & STROMSTEN, S. (2000). Learning context-free grammars with a simplicity bias. In: *Proceedings of the Eleventh European Conference on Machine Learning*, pp. 220–228. Barcelona: Springer-Verlag.

MACWHINNEY, B., ed. (1999). *The emergence of language*. Mahwah, NJ: Lawrence Erlbaum Associates.

NIYOGI, P. (1998). *The informational complexity of learning*. Boston, MA: Kluwer.

NIYOGI, P. & BERWICK, R. C. (1995). The logical problem of language change. Tech. rep., M.I.T.

NOWAK, M. A., KOMAROVA, N. & NIYOGI, P. (2001). Evolution of universal grammar. *Science* **291**, 114–118.

STEELS, L. (1999). The puzzle of language evolution. *Kognitionswissenschaft* **8**.

STOLCKE, A. (1994). *Bayesian Learning of Probabilistic Language Models*. Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, University of California at Berkeley.

WEXLER, K. & CULICOVER, P. (1980). *Formal principles of language acquisition*. Cambridge MA: MIT Press.

WOLFF, J. G. (1982). Language acquisition, data compression and generalization. *Language & Communication* **2**, 57–89.

YANG, C. D. (2000). Internal and external forces in language change. *Language Variation and Change* **12**, 231–250.

VAN ZAANEN, M. (2001). *Bootstrapping Structure into Language: Alignment-Based Learning*. Ph.D. thesis, School of Computing, University of Leeds.

ZUIDEMA, W. (2003). How the poverty of the stimulus solves the poverty of the stimulus. In: *Advances in Neural Information Processing Systems 15 (Proceedings of NIPS'02)* (Becker, S., Thrun, S. & Obermayer, K., eds.). Cambridge, MA: MIT Press. (forthcoming).

A Memory-less learner and batch learner

To estimate the probability that memory-less learning finds the correct grammar after a certain number (b) of sample sentences, we need to consider the inverse: the probability that the algorithm still has a wrong hypothesis after b sample sentence.

$$P(right\ grammar\ after\ b\ samples) = 1 - P(wrong\ grammar\ after\ b\ samples) \tag{1}$$

The probability that the learner still has the wrong hypothesis, depends on the probability that it initially chose the wrong hypothesis (simply $(N-1)/N$) times the probability that it remained for all b sentences at a wrong hypothesis. If it makes no essential difference which wrong grammar is the present hypothesis and how long it has held it as the hypothesis[5], the probability that the algorithm remain for b sentences at a wrong hypothesis is simply $P(remain)^b$. Hence,

$$P(wrong\ grammar\ after\ b\ samples) = \frac{(N-1)}{N}\left(P(remain)\right)^b \tag{2}$$

The probability to remain at a wrong grammar for each random input sentence is given by the probability that that input sentence happens to be consistent with the present (wrong) grammar, plus the probability that the algorithm jumps to another wrong grammar:

$$P(remain) = P(consistent) + P(another\ wrong\ grammar) \tag{3}$$

The probability that a sentence is consistent with a wrong grammar is simply the similarity parameter a in Nowak *et al.* (2001). The probability that the algorithm jumps to another wrong grammar is given by the probability that the input sentence is inconsistent $(1-a)$ times the fraction of other wrong grammars $((N-2)/N)$.

Putting all this together, the probability (q) that the memory-less learner has found the correct grammar after b input sentences is given by (Komarova *et al.*, 2001)[6]:

$$\begin{aligned} q_{memoryless} &= 1 - \frac{(N-1)}{N}\left(a + \frac{(N-2)(1-a)}{N-1}\right)^b \\ &= 1 - \frac{(N-1)}{N}\left(1 - \frac{(1-a)}{N-1}\right)^b \end{aligned} \tag{4}$$

The probability that the batch learner has found the correct grammar after b input sentences is found by Nowak *et al.* (2001) to be

$$q_{batch} = \frac{\left(1 - \left(1 - a^b\right)^N\right)}{(Na^b)} \tag{5}$$

[5]That is the case, for the memory-less learner, under the assumption of Nowak *et al.* (2001) that all grammars are equally similar to each other. In contrast, in a Principles & Parameters model, we can calculate the expected similarity based on estimates of how many parameters are revealed in a single sentence. Under the assumption that every sentence reveals m parameters, that all parameters are Boolean and that all parameters are revealed with equal probability: $a \approx \left(\frac{1}{2}\right)^m$. $a \approx \left(\frac{1}{2}\right)^m$. a is then an expected value rather than a constant, and equation (2) needs to be adapted. For simplicity, we will here follow the assumption of Nowak et al.

[6]Note that there is an error in this equation in Nowak *et al.* (2001) that is corrected in Komarova *et al.* (2001)

Language adaptation helps language acquisition

Willem Zuidema

Artificial Intelligence Laboratory
Vrije Universiteit Brussel
Pleinlaan 2, B-1050, BRUSSELS, Belgium
jelle@arti.vub.ac.be

I evaluate in a computational simulation two mathematical models that are interpreted as support for the argument from the poverty of stimulus, and for the view that underlying the human language capacity is an innate, symbolic and language-specific Universal Grammar. The model I present consists of an evolving population of language learners, that learn a grammar from their parents and get offspring proportional to their success in communicating with other individuals in their generation. In the simulations I observe – contrary to the predictions from work in the Universal Grammar tradition – successful acquisition of "unlearnable" grammars and grammatical coherence in the population. The reason for these surprising results is that language acquisition is a very particular type of learning problem: *language learners learn language from language learners*, i.e. the target of the learning process is itself the outcome of a learning process. That opens up the possibility of language itself to adapt to the language acquisition procedure of children, because learners are only presented with targets that other learners have been able to learn, and most likely even with targets that are best learnable.

Human language is one of the most intriguing adaptive behaviors that has emerged in evolution. With our language we can communicate about events that have happened in the past or will happen in the future; we can express complex causal relations, phrase questions or imperatives, and share in detail previous experiences. Language makes it possible to express an unbounded number of different messages, and it serves as the vehicle for transmitting knowledge that is acquired over many generations. Human language, with its grammar, is therefore viewed as the last *major transition* in evolution (Maynard-Smith and Szathmáry, 1995): a transition that opens up a fundamentally new level of information transmission and selection.

Not surprisingly, the origins of language are a central issue in both evolutionary biology and the cognitive sciences. The dominant explanation for the origins and nature of human language postulates a "Universal Grammar": an innate system of principles and parameters, that is universal, genetically specified and independent from other cognitive abilities. In an influential paper, Pinker and Bloom (1990) have argued that the only

viable explanation for the origins of this Universal Grammar is the process of natural selection.

In this paper, I study an argument that lies at the heart of this dominant position: the argument from the poverty of stimulus. This argument states that children have insufficient evidence to learn the language of their parents without innate knowledge about which languages are possible and which are not. This claim is backed-up with a series of mathematical models. I evaluate two such models in the context of a computational simulation, and arrive at very different conclusions. Following other researchers in the field of artificial life and simulation of adaptive behavior that have studied the origins of communication and human language (e.g. MacLennan and Burghardt, 1993; Steels, 1996) I have made fruitful use of some of their strong points: the emphasis on situatedness, self-organization and computational modeling.

The model that I will present is a "language game" model, with the important characteristic that it models a *population* of agents. In such models, *learners learn from learners*, and the target of learning is therefore not fixed, like in traditional machine learning problems, but is itself the outcome of the dynamics. This makes it possible that the language adapts to the properties of the learners. My goal is to study this phenomenon in the context of a precise model. I will demonstrate that it is real and that it has important consequences for the traditional argument from the poverty of stimulus.

The argument from the poverty of stimulus

In the Universal Grammar tradition, phrase structure grammars (or rewriting grammars) are the usual representation to describe the syntactical form, the "phrase structure", of language. They consist of a list of rules, that each rewrite some sequence of symbols to another sequence of symbols. Symbols are elements of a terminal alphabet or a non-terminal alphabet. For our purposes, we can think of terminal symbols as the words of a language, whereas non-terminal symbols are necessary intermediate states in generating sentences. There is one special non-terminal symbol S, called the start symbol. Language production, "derivation", with this formalism, is applying the rules iteratively to the start symbol until some sequence of terminal symbols is reached. Language comprehension, "parsing", on the other hand, is, given such a terminal string, finding a series of rewriting steps that lead from S to this terminal string. The following grammar can, for instance, generate the English sentences "the child cries" or "the child laughs":

$$\begin{array}{rclcrcl} S & \mapsto & NP\ VP & \qquad & VP & \mapsto & cries \\ NP & \mapsto & the\ child & & VP & \mapsto & laughs \end{array}$$

Phrase-structure grammars are in many ways too powerful as a representation for linguistic competence. Linguists therefore assume that the class of human languages corresponds only to a small subset of the set of all phrase-structure grammars. This is accomplished by assuming a set of principles that restrict the possible actual rules of the grammar, and a set of parameters that determine which are the actual rules that

are chosen. Language acquisition in this tradition is reduced to determining the right set of parameters. The principles, the parameters and the learning algorithm ("language acquisition device") are together termed Universal Grammar.

For many linguists, Universal Grammar is the solution for two phenomena that would otherwise be difficult to explain: (i) the rapid acquisition of language by infants from very limited training data; (ii) the universal tendencies in the structure of languages. Children learn language remarkably fast, from very few examples and without hardly any negative feedback. Children do show in their speech knowledge of all kinds of specific grammatical rules, without ever having encountered examples of those rules. The only explanation, according to this theory, is that these aspects of language are innate. This is known as the "argument from the poverty of stimulus".

Gold (1967) provides the most well-known formal basis to this argument. Gold introduced the criterion "identification in the limit" for evaluating the success of a supervised learning algorithm: with an infinite number of training samples all hypotheses of the algorithm should be identical, and equivalent to the target. Gold showed that context-free grammars are in general not learnable by this criterion from positive samples alone. This proof is based on the fact that if one has a grammar G that is consistent with all the training data, one can always construct a grammar G' that is slightly more general: i.e. the language of G, $L(G)$ is a subset of $L(G')$. Because natural languages are thought to be at least as complex as context-free grammars, and negative feedback is assumed to be absent, Gold's analysis is usually interpreted as strong support for the argument from the poverty of stimulus.

Nowak et al. (2001) provide a novel variant of the argument from the poverty of stimulus, that is based on a mathematical model of the evolution of grammars. They introduce an elegant formalism in which they assume that there is a finite number of possible grammars. Further, they assume that newcomers (infants) learn their grammar from the population, where more successful grammars have a higher probability to be learned and mistakes are made in learning. The system can now be described in terms of the changes in the relative frequencies x_i of each grammar type i in the population.

The first result that Nowak et al. obtain is a "coherence threshold". This threshold is the necessary condition for grammatical coherence in a population, i.e. for a majority of individuals to use the same grammar. They show that this coherence depends on the chances that a child has to correctly acquire its parents' grammar. This probability is described with the parameter q. Nowak et al. show mathematically that there is a minimum value for q to keep coherence in the population. If q is lower than this value (called q_0), all possible grammar types are equally frequent in the population and the communicative success in minimal. If q is higher than this value, one grammar type is dominant; the success is much higher than before and reaches 100% if $q = 1$.

The second result relates this required fidelity to a lower bound (b_0) on the number of sample sentences that a child needs. Nowak et al. make the crucial assumption that each language is equally expressive and dissimilar from any other language. With that assumption they can show that b_0 is proportional to the total number of possible grammars N. Of course, the number of sample sentences b is finite. Nowak et al. conclude that only

if N is relatively small can a stable grammar emerge in a population. I.e. the population dynamics require a restrictive Universal Grammar.

Language Adaptation

The models discussed above have in common that they implicitly assume that every possible grammar is equally likely to become the target grammar for learning. If even the best possible learning algorithm cannot learn such a grammar, the set of allowed grammars must be restricted. There is, however, reason to believe that this assumption is not the most useful for language learning. Language learning is a very particular type of learning problem, because the outcome of the learning process at one generation is the input for the next. Simon Kirby has coined models of this phenomenon "Iterated Learning Models".

A seminal paper that illustrates the consequences of this fact is Kirby (2000). He studied the language systems that arise in a simulation with agents that have quite sophisticated learning abilities and communicate about situations in a structured "meaning space". The tasks of the agents is to acquire a grammar such that they best understand the utterances of other agents. The training samples are form–meaning pairs; agents in Kirby's model thus have direct access to the "intentions" of the speaker. The "meanings" are expressions in a predicate logic, that shows an explicit combinatorics (e.g. "John finds Mary" vs. "John sees Mary"). Nevertheless, grammars can be non-compositional to convey such combinatorial meanings. Kirby finds in his simulations that after a period in which the grammars are primarily non-compositional and have low expressiveness, a turbulent regime occurs with on average much more expressive grammars, culminating in a stable regime were agents reach maximum expressiveness with rather small grammars. The final solution is a highly structured, compositional grammar with a regular correspondence between meaning elements (e.g. "John") and forms (e.g. "da").

Agents in the model are not restricted to use compositional grammars. Yet, by simply observing each other's behavior, learn from it and occasionally inventing new behaviors, syntactic language emerges. Kirby explains this phenomenon with the observation that the units of I-language (the internal knowledge of language), in the model the rewriting rules, only "survive" if they are faithfully preserved in the mapping to E-language (the external utterances in the "arena of use") and back to another individual's I-language (see also Hurford, 2002). Kirby's model is a nice illustration of the idea that in a population of *learners*, culturally transmitted units in some sense compete for survival, that is for *being learned*. Rewriting rules themselves can become units of selection, and if there is variation, undergo *cultural evolution*. Compositional rules have a selective advantage in such a process, because they can be used to express many different messages and hence tend to be used more often. In another paper (Kirby, 2002), Kirby shows similar results for the emergence of recursive grammars, with a recursively structured meaning space. In this paper I explore the consequences of the phenomenon of language adaptation that Kirby and others have identified (Deacon, 1997; Batali, 1998; Kirby, 2000, 2002; Hurford,

2002) for the arguments for Universal Grammar discussed above.

Model design

In this section I will first discuss some general consideration in the design of the model, and then some specific design choices for the formal *representation* to describe the linguistic abilities of agents, the *learning* algorithm for language acquisition and the population structure that determines the *transmission* of language between agents. These three topics correspond roughly with "representation bias" and "search bias", the two relevant dimensions for any machine learning algorithm (Mitchell, 1997), and "collective dynamics", a dimension unique for multi-agent learning.

I apply some popular themes from artificial life to the language acquisition problem: (i) emphasis on *situatedness*, the observation that the agent's embodiment and environment, including other agents, should form an integral part of the explanation of behavior. Embodiment does not play a major role in this study, but viewing an agent as part of an environment with other agents is crucial; (ii) emphasis on *self-organization*, the phenomenon that complex, global patterns are formed by many simple, local interactions; and (iii) the use of *computational modeling*, an approach to scientific research where assumptions are made explicit, and the behavior of the model is evaluated experimentally. For the purpose of this paper – countering the arguments based on the models discussed above – the clarity of the model is more important than the cognitive plausibility of the algorithms and representations. Cognitive plausibility has therefore been a minor concern.

The main question I try to answer is *Does the language adapt to the bias of the learning algorithm and does it thus become easier to learn?* In order to answer that question I have operationalized "learnability" as the success in communicating with one's parent (the target language) after a fixed number of sample sentences. I use this measure, because both its implementation and its cognitive interpretation are straightforward. It is, however, different from Gold's "identification in the limit" , i.e. 100% accuracy if the the number of samples goes to infinity as the *binary* criterion for learnability. Although I question the empirical relevance of this criterion (observations on historical language change clearly show that there is no "identification" of grammars), I will also look at my results from that perspective.

Representation

I use context-free grammars to represent the linguistic abilities. In particular, the representation is limited to grammars G where all rules are of one of the following forms: (1) $A \mapsto t$, (2) $A \mapsto BC$, or (3) $A \mapsto Bt$. The nonterminals A, B, C are elements of the non-terminal alphabet V_{nt}, which includes the start symbol S. t is a string of terminal symbols from the terminal alphabet V_t. Note that each rule of the third type could be replaced by one rule of the first type and one rule of the second type (e.g. $S \mapsto Xq$ can be replaced by $S \mapsto XY$ and $Y \mapsto q$). The grammar would then be in Chomsky Normal

Form. Since every context-free grammar can be transformed to that form, the restrictions on the rule-types above do not limit the scope of languages that can be represented. They are, however, relevant for the language acquisition algorithm that will be discussed below. Note that the class of languages that my formalism can represent is unlearnable by Gold's criterion (Gold, 1967). I.e. there will always be multiple grammars that are consistent with the training data, such that the target grammar can not be uniquely identified.

For determining the language L of a certain grammar G I use simple depth-first exhaustive search of the derivation tree. For computational reasons, the depth of the search is limited to a certain depth d, and the string length is limited to length l. The set of sentences ($L' \subset L$) used in training and in communication is therefore finite (and strictly speaking not a a context-free, but a regular language). In the communication between two agents, the speakers chooses a random element s of its language, and the hearer checks if s is an element of its own language. If so, the interaction is a success, otherwise it is a failure.

Learning

The language acquisition algorithm used in the model consists of three operations: (i) incorporation, (ii) compression and (iii) generalization. The learner learns from a set of sample strings (sentences) that are provided by a teacher. The design of the learning algorithm is inspired by Kirby (2000), but leaves out all semantics. Langley and Stromsten (2000) describe, in a very different context, a very similar algorithm.

Incorporation: *extend the language, such that it includes the encountered string*;

Compression: *substitute frequent and long substrings with a nonterminal, such that the grammar becomes smaller and the language remains unchanged*;

Generalization: *equate two nonterminals, such that the grammar becomes smaller and the language larger*;

For the grammar acquisition algorithm these three operations can be used in several set-ups. For the purposes of this paper, I have chosen for simple *off-line learning*: compression and generalization occur after all training strings have been received. To avoid insufficient expressiveness, the generalization phase concludes with a check for the size of the language. If this size is smaller than some minimum, $size(L) < M$, I generate $M - size(L)$ random strings[1] and `incorporate` them in the grammar. This procedure can be considered a substitution for the semantics that is left out in the model.

Transmission

I consider two different transmission schemes. The first is simply iterated learning: agents in a *chain* learn from the previous agent and teach to the next. A simulation runs P of

[1] These strings have a maximum size l_0 which is an important parameter in the results section ($l_0 \leq l$)

such chains in parallel. One can consider this a population, where every generation has P members. Every individual gets exactly one child, and the child learns the language from its parent. All parents are then removed, and the children become parents for a new generation.

The second scheme involves fitness proportional selection. As in Nowak et al. (2001) the fitness f of an agent is determined by its success in communicating with the agents of its own generation. The expected number of offspring is proportional to this fitness. The population size is constant (P). The difference with the chain condition is thus that a parent can also have no child or several children, depending on the communicative success *within* its own generation.

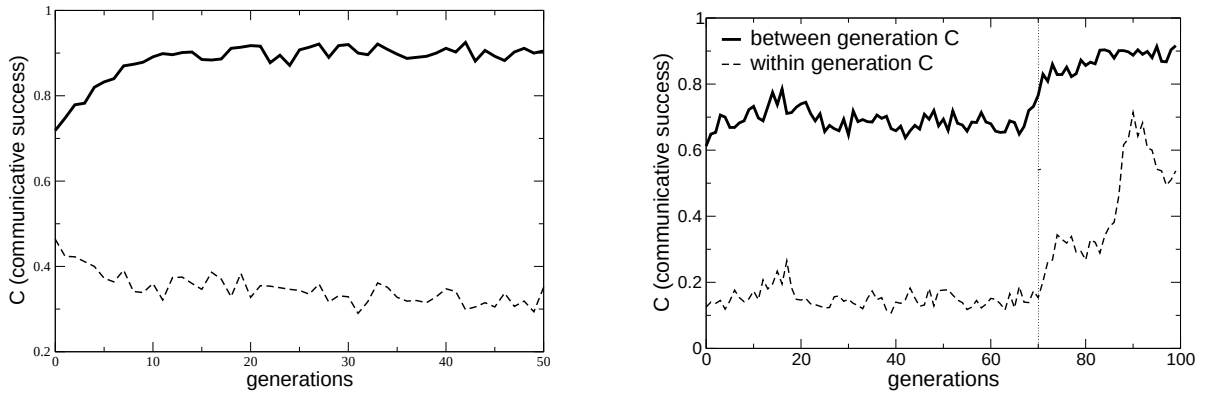
Results

I have implemented the model in C++ and run many experiments. The results are shown in figures 1a and 1b. Both graphs show two curves: (i) the average communicative success of speaking with their parent which is the measure for the *learnability* of the language (labeled "between generation C"), and (ii) the average communicative success of agents speaking with other agents of the same generation (labeled "within generation C") which gives the fitness (f_i) of agents and is a measure for the grammatical variation in the population.

Figure shows a partial reproduction of the results of (Kirby, 2000). In this simulation, under the chain condition and thus without any selection, the between generation communicative success rises steadily from a low value (just over 0.7) to a high value (0.9). In the initial stage the grammar shows no structure, and consequently almost every string that the grammar produces is idiosyncratic. A child in this stage typically hears strings like "030", "3302", "0310", "1213", or "2320" from its parent. It can not discover many regularities in these strings. The child therefore can not do much better than simply reproduce the 70 or so different strings it heard (i.e. 100 random draws from 100 different strings), and generate around 30 random new strings to make sure its language obeys the minimum of 100 strings.

However, in these randomly generated strings, sometimes regularities appear. I.e., a parent may use the randomly generated strings "3202", "1202", "2002" and "3002". When this happens the child tends to analyze these strings as different combinations with the building block "02". Thus, typically, the learning algorithm generates a grammar with the rules $S \mapsto 32a$, $S \mapsto 12a$, $S \mapsto 20a$, $S \mapsto 30a$, and $a \mapsto 02$. When this happens to another set of strings as well, say with a new rule $b \mapsto 1$, the generalization procedure can decide to equate the non-terminals a and b. The resulting grammar can then generalize from the observed strings, to the unobserved strings "321", "121", "201" and "301". The child still needs to generate random new strings to reach the minimum, but fewer than in the case considered above.

The interesting aspect of this becomes clear when we consider the next step in the simulation, when the child becomes itself the parent of a new child. This child is now



(a) chain condition

(b) fitness proportional selection

Figure 1: (a) Results from a run under the chain condition. The results show a steady growth of the between generation C from 0.7 to about 0.9. A child receives about 70% of the parent's language as sample data. The language has thus become considerably better learnable than a random language is. The within generation C (which plays no role in the dynamics, as there is no selection) goes down from around 0.5 (chance level for random language with maximum initial string length $l_0 = 4$), to around 0.3. Parameters are: $V_t = \{0, 1, 2, 3\}$, $V_{nt} = \{S, a, b, c, d, e, f\}$, P=20, T=100, M=100, l_0=4. (b) Results from a run under fitness proportional selection. This figure shows that there are regions of grammar space where the dynamics are apparently under the "coherence threshold" (Nowak et al., 2001), while there are other regions where the dynamics are above this threshold. The parameters, including the number of sample sentences T, are still the same, but the language has adapted itself to the **bias** of the learning algorithm. Parameters are: $V_t = \{0, 1, 2, 3\}$, $V_{nt} = \{S, a, b, c, d, e, f\}$, P=20, T=100, M=100, l_0=12

presented with a language with more regularities than before, and has a fair chance of *correctly* generalizing to unseen examples. If, for instance, it only sees the strings "3202", "1202", "2002", "121", "201" and "301", it can, through the same procedure as above, infer that "3002" and "321" are also part of the target language. This means that (i) the child shares more strings with its parent than just the 70 it observes and consequently shows a higher between generation communicative success, and (ii) regularities that appear in the language by chance, have a fair chance to remain in the language. In the process of iterated learning, languages can thus become more structured and better learnable.

Very similar results with different formalisms were already reported before (e.g. Kirby, 2000; Brighton, 2002), but here I have used context-free grammars and the results are therefore directly relevant for the interpretation of Gold (1967). When we study this effect in a version of the model where *selection* does play a role, it is also relevant for the analysis Nowak et al. (2001). Whereas in the simulations of figures the target languages has been relatively easy (the initial string length is short, i.e. 4), here the learning problem is very difficult (initial string length is long, i.e. 12). For a long period the learning is therefore

not very successful, but around generation 70 the success suddenly rises. With always the same T (number of sample sentences), and with always the same grammar space, there are regions of that space where the dynamics are apparently under the "coherence threshold" (Nowak et al., 2001), while there are other regions where the dynamics are above this threshold. The language has adapted to the learning algorithm, and, consequently, the coherence in the population does not satisfy the prediction of Nowak et al.

In many runs (not shown here) I have also observed 100% learning accuracy of children. This typically occurs under parameter settings where the "poverty of stimulus" is not very severe. E.g. when the number of training samples (T) is twice the required minimum expressiveness (M). The grammars that emerge in all simulations where the between-generation C grows to non-trivial levels (i.e. above 70% for the parameters I have used here) are combinatorial. In some, but not all cases, they are also recursive[2].

Discussion

The widely held intuition in linguistics is that the grammars of natural languages are not learnable from available evidence, and this intuition is supported with a series of mathematical models. In such studies one derives from the properties of the learning procedure (the "procedural bias"), fundamental constraints on the nature of the target grammar. These constraints are thought to come from an innate predisposition: the "Universal Grammar", that provides strict, a-priori constraints on the set of grammars a child needs to consider in the first place (the "representation bias").

The underlying assumptions of these "proofs" of Universal Grammar have been controversial, and I suspect that some of this criticism is in fact justified. However, I leave that issue here as an empirical question and have in stead made similar assumptions as in the mathematical models, but still arrive at very different conclusions: I do observe successful acquisition of grammars that are unlearnable by Gold's criterion. Further, I observe grammatical coherence although many more grammars are allowed in principle than Nowak et al. calculate as an upper bound. The reason for these surprising results is that language acquisition is a very particular type of learning problem: it is a problem where the target of the learning process is itself the outcome of a learning process. That opens up the possibility of language itself to adapt to the language acquisition procedure of children. In such iterated learning situations (Kirby, 2000), learners are only presented with targets that other learners have been able to learn.

Isn't this Universal Grammar in disguise? Learnability is – consistent with the undisputed proof of Gold (1967) – still achieved by constraining the set of targets. However, unlike in usual *interpretations* of this proof, these constraints are not strict (some grammars are better learnable than others, allowing for an infinite "Grammar Universe"), and

[2]The lack of recursive grammars in some simulations is not a problem for the conclusion that the language adapts to the learning algorithm; it is, however, a problem for projecting this phenomenon the hypothesized course of the evolution of human language, because recursion is seen as one of human language's fundamental characteristics. Kirby (2002), in a model that includes semantics, has already shown similar results with recursive syntax.

they are not a-priori: they are the outcome of iterated learning. The poverty of stimulus is now no longer a problem; in stead, the ancestors' poverty is the solution for the child's.

Acknowledgments This work builds on previous work that was done in close collaboration with Paulien Hogeweg. I thank her and my colleagues at the AI-Laboratory in Brussels for valuable hints, questions and remarks. I am funded through the Concerted Research Action fund (G.O.A.) of the Flemish Government and the VUB.

References

Batali, J. (1998). Computational simulations of the emergence of grammar. In Hurford, J. and Studdert-Kennedy, M., (Eds.), *Approaches to the evolution of language: social and cognitive bases*. Cambridge University Press.

Brighton, H. (2002). Compositional syntax from cultural transmission. *Artificial Life*, 8(1).

Deacon, T. (1997). *Symbolic species, the co-evolution of language and the human brain*. The Penguin Press.

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

Hurford, J. R. (2002). Expression / induction models of language. In Briscoe, T., (Ed.), *Linguistic Evolution through Language Acquisition: Formal and Computational Models*. Cambridge University Press.

Kirby, S. (2000). Syntax without natural selection: How compositionality emerges from vocabulary in a population of learners. In Knight, C., Hurford, J., and Studdert-Kennedy, M., (Eds.), *The Evolutionary Emergence of Language: Social function and the origins of linguistic form*. Cambridge University Press.

Kirby, S. (2002). Learning, bottlenecks and the evolution of recursive syntax. In Briscoe, T., (Ed.), *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press.

Langley, P. and Stromsten, S. (2000). Learning context-free grammars with a simplicity bias. In *Proceedings of the Eleventh European Conference on Machine Learning*, pages 220–228. Barcelona: Springer-Verlag.

MacLennan, B. J. and Burghardt, G. M. (1993). Synthetic ethology and the evolution of cooperative communication. *Adaptive Behavior*, 2(2):161–188.

Maynard-Smith, J. and Szathmáry, E. (1995). *The major transitions in evolution*. Morgan-Freeman.

Mitchell, T. (1997). *Machine learning*. McGraw-Hill.

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

Pinker, S. and Bloom, P. (1990). Natural language and natural selection. *Behavioral and brain sciences*, 13:707–784.

Steels, L. (1996). A self-organizing spatial vocabulary. *Artificial Life Journal*, 2(3).

The evolutionary biology of language

Jelle Zuidema
Institute for Logic, Language and Computation
University of Amsterdam
jzuidema@science.uva.nl

Unpublished Tutorial Paper, version March 8, 2006

Abstract

What are the requirements for scenarios of the biological evolution of language? In this paper I survey a number of simple but fundamental models from population genetics, evolutionary game-theory and social evolution theory. This review yields a list of required elements of evolutionary explanations in general, and of explanations for language and communication in particular.

1 Introduction

There are two distinct ways in which the study of evolution and the study of natural language overlap. First, they overlap in the search for an evolutionary explanation for why humans, and humans alone, are capable of acquiring and using natural languages. Second, the process of evolution in biology and the historical process of language change bear many similarities, and these parallels have played a role in the development of theories in both fields since the time of Darwin. I will throughout this paper refer to these issues as the *biological evolution of language* (or "the language faculty") and the *cultural evolution of language(s)* respectively.

Both issues have received a great deal of attention in recent years, leading to a plethora of theories and models (Hurford *et al.*, 1998; Christiansen & Kirby, 2003). Many proposals involve a single mechanism or factor responsible for the emergence of modern natural languages. In some cases, extensive scenarios for the evolution of language are proposed. Although this enormous body of work contains a great number of interesting ideas and findings, there are also a number methodological problems. First, it is extremely difficult to relate separate proposals to each other, because of a lack of consensus on terminology and basic assumptions. Second, it is extremely difficult to evaluate the internal consistency and empirical validity of proposed theories, because of a lack of formal rigor.

In some ways this situation is reminiscent of the state of the whole field of evolutionary biology before the establishment of theoretical population genetics by Fisher, Wright, Haldane and others in the 1920s and 30s. Their mathematical models, and the subsequent informal "modern synthesis", convinced biologists of the central role of natural selection in evolution. Confusion remained about the units of selection, but with the settling of the group selection debate by Maynard Smith (1964) and Williams (1966) a relative consensus emerged about the minimum requirements for evolutionary explanations, as well as a common vocabulary in which disagreements can be phrased. In the interdisciplinary field of language evolution, this clarity is yet lacking. In this paper, I will review some simple mathematical models from evolutionary biology, and evaluate how they can be applied to both the biological and the cultural evolution of language.

I will start with some classical results from population genetics, about the way gene frequencies in a population change as a result of mutation and selection, and then discuss the case for viewing natural selection as optimisation, as well as the problems with this view. This optimisation view then provides a natural bridge to evolutionary game theory, where the targets of optimisation shift because the opponents in the game evolve as well. Finally, extensions to social evolution models that deal with kin selection, will lead us to the issue of levels of selection, and clarify the relation of cultural evolution models – with the dynamics happening at the level of cultural replicators – to evolutionary biology generally.

2 Adaptation for Language

When chimpanzees, our closest living relatives, are taught human language, they acquire several hundreds of signals (Gardner & Gardner, 1969; Savage-Rumbaugh *et al.*, 1986). They fail, however, to produce speech sounds themselves, to acquire the many tens of thousands of words in natural languages, and to grasp the use of even the most basic rules of grammar (Terrace, 1979). Human infants, in contrast, acquire their native language rapidly. They produce speech sounds

and comprehend simple words before the age of 1, produce their first words soon after their first birthday and the first grammatical constructions before their second birthday (Tomasello & Bates, 2001).

Why? Clearly there is something special about humans that makes them extra-ordinarily apt to acquire and use natural languages. Among other things, the anatomy of the vocal tract, the control mechanism in the brain for complex articulation and the cognitive ability to analyse and produce hierarchically structured sentences appear to be qualitatively different in humans than in other apes. But not only humans are special; there is also something special about natural languages that makes them extra-ordinarily apt to be acquired and used by humans.

How did this tight fit come about? One possibility is that the human capacity for language has emerged purely as a side-effect of the many changes in anatomy and cognition that occurred in the hominid lineage. The tight fit itself, in such a scenario, doesn't need to be accidental, because a cultural evolution scenario predicts that language will adapt to the peculiar biophysical and cognitive features of humans that themselves have evolved for other reasons.

Although this possibility cannot be dismissed, from a biological point of view it does not appear very likely. Humans spend around 3 hours a day or over 20% of their awake time talking (Dunbar, 1998, and references therein), verbal abilities play a significant role in social status and, it seems, in both the reproductive success of individuals and the success of our species as a whole. Such a salient characteristic of any organism would require a Darwinian, evolutionary explanation. Hence, although the side-effect option is a possibility, it can only be the conclusion of an elaborate investigation, and not serve as null hypothesis. Nevertheless, although language as a whole might be considered a biological adaptation, many specifics about language (language universals) are perhaps better understood as the outcome of cultural evolution. In this view, the complex results of cultural evolution and social learning have had indirect consequences for biological evolution.

If we want to investigate specific hypotheses on adaptations for language, what form should such hypotheses take? The early formal models in population genetics are a useful starting point. But first, it should be clear that any statement about biological evolution is a statement about how genes mutate and spread in a population through random drift and selection. That statement in no way reflects the form of genetic determinism or naivety about "language genes" that have made some evolutionary linguists wary to talk about genes at all. But if properties of language are to be explained by some biological endowment, which in turn is to be explained as an adaptation for language, then we need to be explicit and postulate a series of altered genes that influence the ability for language. Such genes can have many additional non-linguistic effects (an illustrative example is the recently discovered FOXP2 gene, that, when mutated, causes a range of problems in language processing as well as in sequencing orofacial movements, Lai *et al.* 2001). We can phrase this requirement[1] as follows:

Requirement 1 (Heritability) *Evolutionary explanations for the origins of a trait need to postulate genetic changes required for that trait.*

3 Evolution as Gene Frequency Change

A formal model of evolution as gene frequency change can be built-up in the following way. Consider first that in humans, as in almost all multicellular organisms, every individual inherits two sets of genes, one from the father and one from the mother. If there is to be any change, we need to consider at least two different variants, alleles, for each gene locus, and monitor the increase in frequency of one allele at the expense of the other. In figure 1 the Mendelian model of inheritance of two alleles – A and a at a single locus – is depicted. Adults (top row) have a genome that is of any of the three possible types AA, Aa or aa (Aa and aA are equivalent). These adults produce sperm and egg-cells (second row) with just a single copy of the gene under consideration. In sexual reproduction, a sperm-cell and an egg-cell fuse, and grow out to a new individual (third row). Evolution, in this simple scheme, concerns the change in frequencies of the genotypes AA, Aa or aa, or the change in frequencies of the alleles A and a.

The Hardy-Weinberg model (developed independently by British mathematician Godfrey Harold Hardy, 1908 and German physician Wilhelm Weinberg, 1908; see Crow, 1999) describes the gene frequencies if there is no mutation or selection. Consider the frequencies of the three genotypes (top row) at any particular point in time, and call these frequencies D, H and R. The frequencies of the alleles A and a in the sperm and egg-cells are simply:

$$\begin{aligned} \textit{frequency of } A\text{: } p &= D + \frac{1}{2}H \\ \textit{frequency of } a\text{: } q &= R + \frac{1}{2}H, \end{aligned} \qquad (1)$$

because individuals with genotype AA or aa will always pass on an A or a respectively to their sperm and

[1] Of course, one can sensibly study the evolution of traits for which the genetic component has not been identified. The point here is to emphasise that biological evolution implies genetic changes. The "requirements" in this paper concern the ultimate evolutionary explanation for a trait; of course, not every evolutionary model study will be able to meet all requirements.

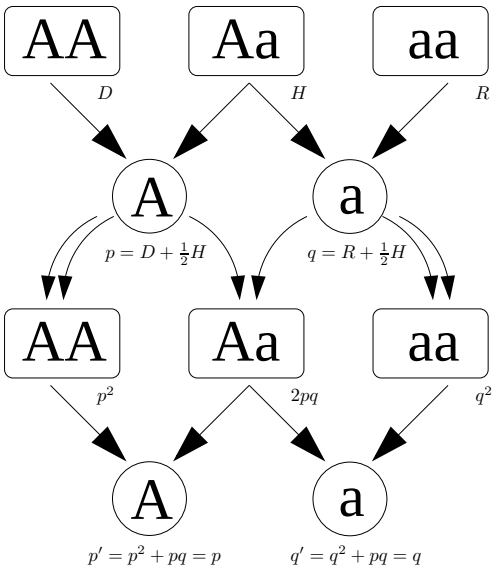


Figure 1: Mendel's model of inheritance, and the Hardy-Weinberg model of allele and genome frequencies under Mendelian inheritance with no selection nor drift.

egg-cells, but individuals with genotype Aa only half of the time.

Under a number simplifying assumptions (including random mating and meiosis, an infinite population and no sex differences at the relevant locus), the frequencies of the three genotypes in the offspring are simply $D' = p^2$, $H' = 2pq$ and $R' = q^2$, because you need two A's or a's to make an AA or aa respectively, and you need an A from either the father or the mother and an a from the other parent to make an Aa. When this offspring then starts producing sperm- and egg-cells, the frequencies of the alleles A and a are:

$$\begin{aligned} new\ frequency\ p' &= D' + \frac{1}{2}H' = p^2 + pq \\ new\ frequency\ q' &= R' + \frac{1}{2}H' = q^2 + pq. \end{aligned} \quad (2)$$

Hardy and Weinberg's simple but fundamental observation is that because $p + q = 1$ (the total frequency of all alleles must be 1, and thus $q = 1 - p$), it follows that p and q are constant under this model of inheritance:

$$p' = p^2 + pq = p^2 + p(1-p) = p^2 + p - p^2 = p. \quad (3)$$

This result shows that under Mendelian inheritance existing variation in gene frequencies is maintained. This is in contrast with "blending inheritance" (the assumed model of inheritance before the rediscovery of Mendel's laws around 1900), where a child's trait values are the average of the parents' and variation quickly dissipates over time. The result played a crucial role in reconciling Mendelian genetics with Darwinian evolutionary theory, because it showed that under reasonably low mutation rates enough variation can be built up for natural selection to operate (Fisher, 1930, chapter 1).

The Hardy-Weinberg model can be extended in a straightforward manner to include the effects of selection. Natural selection, in Darwin's theory, is the consequence of differences in survival rates to the age of reproduction and the differences in reproductive success. These effects can be summarised with a fitness coefficient for each of the possible genotypes, which gives the expected number of offspring. A high coefficient w_{AA} means that individuals of genotype AA live long and reproduce successfully, such that their genes are well represented in the next generation. In terms of the equations, this just requires weighting the contributions of parents of each genotype with the relevant fitness coefficient:

$$p' = \frac{p^2 w_{AA} + pq w_{Aa}}{\overline{w}}, \quad (4)$$

where $\overline{w}$ is the average fitness and given by:

$$\overline{w} = p^2 w_{AA} + 2pq w_{Aa} + q^2 w_{aa} \quad (5)$$

(this term is needed to account for changes in population size due to reproduction and selection).

Equation (4) gives us a first handle on the requirements for evolutionary innovation, and, hence, evolutionary explanations. First of all, natural selection operates on genotypic and phenotypic variation. Second, natural selection favours fitter genes and individuals over less fit ones. Both the variation and the fitness differences need to be made explicit:

Requirement 2 (Strategy set) *Evolutionary explanations need to postulate a set of possible genotypes and phenotypes, as well as the mutations that can move an organism from one genotype–phenotype to another.*

Requirement 3 (Payoff function) *Evolutionary explanations need to postulate a function that relates the possible genotypes–phenotypes in a given environment (that may include other evolving individuals) to fitness.*

If we are interested in a specific biological innovation – that is, a mutation – that was relevant for learning or using language, we need to consider the situations before and after that mutation. In the simplest case, a is the preexisting gene that is initially shared by the whole population, and A is the mutated version of a that has arisen in a single individual. Hence, initially $q \approx 1$ and $p \approx 0$. If A is to play a role in an evolutionary scenario, we need to establish that allele A did start to spread in the population (as sketched in figure 2); in other words, that p increases. We can formulate this requirement as follows:

Requirement 4 (Invasibility) *Innovations in an evolutionary scenario need to be able to* invade *a population; that is, an innovation should spread in a population where it is extremely rare.*

If we know all fitness coefficients, it is straightforward to work out what happens to the frequency of the new mutation. As it turns out A will spread if $w_{Aa} > w_{aa}$, and it will get fixed ($p = 1$) if $w_{AA} > w_{Aa}$. In other words, the fitness of the new gene must be greater than that of the old one, and the new gene must, to some extent, be *dominant* over the old one such that its effects are noticed in individuals that inherit copies of both genes from each of the parents. In fact, the difference in fitness between the two variants must be significant, at least large enough for the new gene not to get lost by chance fluctuations (Fisher, 1922) and to get established after a reasonable number of generations (Haldane, 1932). Note that these results depend on some strong assumptions, including an infinite population with randomly interacting individuals. In small populations with non-random interactions different dynamics can occur.

4 Evolution as Optimisation

Since Darwin (1859), the notion of "adaptation" has played a major role in evolutionary thinking. His work offered a coherent framework to study the traits of organisms in terms of their *function* for survival and reproduction. Even before the mechanisms of genetic inheritance were unravelled, Darwin thus transformed biology from a descriptive to an explanatory science. In the early 1920s the "founding fathers" of population genetics – Fisher, Wright and Haldane – worked out what happens to a single new gene when it appears in a population. But do the dynamics described by equation (4) constitute "adaptation"? In other words, does the predicted change in gene frequencies also mean the population will get better adapted to its environment, i.e. improve its average fitness?

Both Fisher and Wright set out to work out a more general result. I will discuss Fisher's "fundamental theorem of natural selection" (Fisher, 1930) in section 8. Here I will follow Wright's analysis of the average fitness in a population, in particular Roughgarden's (1979) version of these equations. Most mathematical details are in appendix A, but it is useful to look at a couple of Wright's equations. First, it is convenient to look at the *change* in the frequency p at every timestep. This is, using equation (4), given by:

$$\begin{aligned} \Delta p &= p' - p \\ &= \frac{p^2 w_{AA} + pq w_{Aa}}{\overline{w}} - p \end{aligned} \tag{6}$$

This equation can, with a bit of algebra (see equations (25) and (26) in appendix A), be rewritten as follows:

$$\Delta p = \frac{pq}{\overline{w}} \left(p \left(w_{AA} - w_{Aa} \right) - q \left(w_{aa} - w_{Aa} \right) \right) \tag{7}$$

This equation tells us nothing new; it is essentially equation (4) in a different form. However, the new form will prove useful when we have worked out the next equation. We are interested in what happens to the average fitness when the frequency (p) of the innovation changes. Mathematically, that question directly translates into the derivative of $\overline{w}$ with respect to p. The expression for average fitness is given in equation (5). Its derivative, if we assume the fitness coefficients are independent of p and q (that is, no frequency-dependence) turns out to be (as is worked out in equation (23) and (24) of appendix A):

$$\frac{\mathrm{d}\overline{w}}{\mathrm{d}p} = 2 \left(p \left(w_{AA} - w_{Aa} \right) - q \left(w_{aa} - w_{Aa} \right) \right) \tag{8}$$

When we note that equations (7) and (8) are very similar, it is clear that we can replace a large part of (7) with half of (8), and get:

$$\Delta p = \frac{pq}{\overline{w}} \left(\frac{1}{2} \right) \frac{\mathrm{d}\overline{w}}{\mathrm{d}p}. \tag{9}$$

This is a fundamental result for evolutionary biology. The equation says that the change in the frequency of a new gene, will be *in the direction* of the derivative of fitness with respect to that gene's frequency. That means that only if the average fitness increases with increasing p, will the new gene spread. Moreover, the spread will be fastest at intermediate frequencies (high variance) and low average fitness. In other words, evolution – under the assumption mentioned – will act to optimise the average fitness in the population: it will lead to adaptation.

However, the mathematical derivation of this intuitive result also tells us about its limitations. First of all, evolution is shortsighted. We saw a simple example at the end of the previous section: if $w_{Aa} < w_{aa}$ (there is "heterozygous disadvantage"), then the new allele A will not spread in the population, even though at fixation it might improve the mean fitness in the population. Second, evolution needs (heritable) variation. If $pq = 0$, nothing will change. Thirdly, the equation is only valid if the fitness coefficients are *independent* of p and q. That is, whatever the traits are that allele A influences, the usefulness of the innovation should not depend on how many others in the population share it. This condition is obviously violated in the evolution of communication, because the usefulness of a signal will always depend on the presence of others that can perceive and understand it. Fourthly, the original Hardy-Weinberg model brought quite a lot of assumptions, including the independence of the single locus we looked at from other loci, random mating, discrete generations and infinite populations. Some of the consequences of relaxing these and the frequency independence assumptions will be evaluated in the next section.

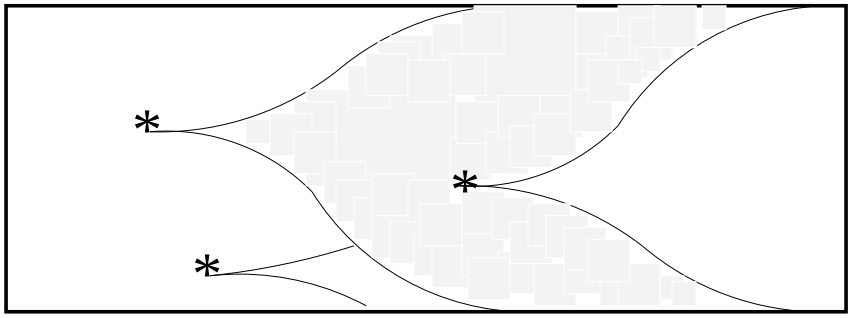


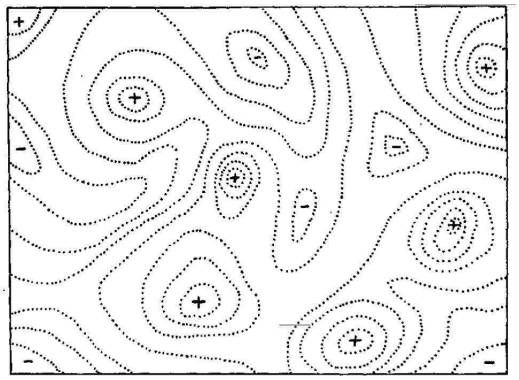
Figure 2: The spread of new genes in a population

Finally, as Fisher (1930) emphasised, these calculations deal only with the direct effects of natural selection. They predict the direction of change, but it is unwarranted to conclude that the average fitness in a population will increase. Environmental conditions might have changed in the mean time and, even if the environment is constant, all individuals in the population are better adapted to it such that competition is fiercer. These effects – not modelled by Wright and Fisher's equations – were collectively labelled "deterioration of the environment" by Fisher.

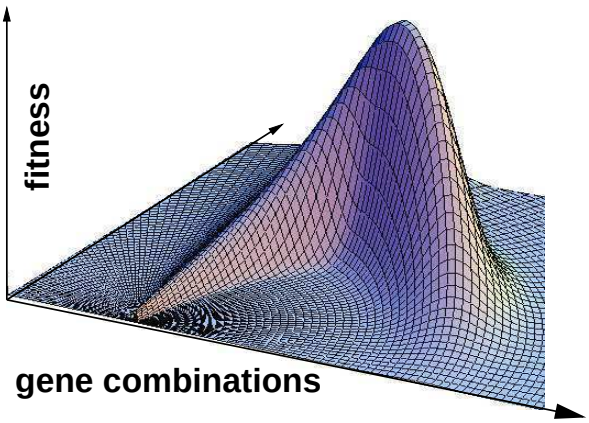
In addition to these quantitative results, Wright made a much more qualitative contribution relating evolution and optimisation. In a paper without any mathematics (Wright, 1932) he introduced an extremely influential metaphor: the **adaptive landscape**. The adaptive landscape is a landscape of 3 or more dimensions, with the plane (or hyperplane) representing the space of possible genotypes, and the height of every point representing fitness (see figure 3). On such a landscape, a population is a collection of points. Mutations correspond to steps in the landscape; selection corresponds to the selective removal of individuals that are lower down. The process of evolution involves the population to climb up-hill, following a local gradient to a local peak.

I will discuss some problems with the concept below. However, the adaptive landscape representation in this form does illustrate Darwin's (1859) insight that for a process of continuing evolution, we need a path of ever increasing fitness from the hypothesised initial point in genotype space to the end result. (In finite populations, stochastic drift can bridge fitness barriers in the adaptive landscape, but only if they are relatively shallow.) For complex traits, such as language, it seems reasonable to postulate a series of many genetic changes. Wright's metaphor highlights the fact that each of these changes needs to confer an adaptive advantage:

Requirement 5 (Fit intermediates) *Explanations for complex traits, that involve a series of genetic changes, need to show a path of fit intermediates, from the hypothesised initial state to the desired end state.*



(a) Wright's graph of the adaptive landscape



(b) A computer-generated 3d adaptive landscape

Figure 3: The adaptive landscape of fitness as a function of genotype. The graphs illustrate hypothetical examples in which two genes have a continuous range of effects. Real organisms have, in contrast, a discrete set of possible genotypes involving many more than two genes. Thus, mutations can take them in very many directions. This high dimensionality makes it more likely that there is some path uphill to the "adaptive peak" (see Provine (1986), chapter 9). (a) is a graph from Wright (1932). The original caption is: "Diagrammatic representation of the field of gene combinations in two dimensions instead of many thousands. Dotted lines represent contours with respect to adaptiveness." (b) is taken from Barton & Zuidema (2003).

This requirement is important, but it might not be as problematic as it looks at first sight. First, although evolution will generally lead uphill, there is some room for random processes as well. Wright used the adaptive landscape metaphor to explain the effects of increases or decreases of the rate of mutation and the strength of selection. He also discussed at some length the effects of small population sizes, where inbreeding will lead to the non-selective process of genetic drift: random deviations from the locally optimal genotype due to accumulation of mutations and a lack of variation for selection to operate on. Wright's shifting balance theory (or at least one version of it) argues that the additional variation inherent in subdivided and inbreeding populations could help the population as a whole bridge fitness barriers. Although the shifting balance theory has little empirical support (Coyne, Barton & Turelli, 2000), the basic idea that, under some conditions, genetic drift could help bridge a fitness barrier remains.

Second, one of the basic tenets of evolutionary biology is that all life originates from the same source. If that is true, all complex traits of all organisms are connected through paths of fit intermediates. Thus, if we wonder if there is a path on the adaptive landscape through which humans could evolve wings, the answer must be yes. Humans, bats and birds have a common ancestor, so there must be at least one series of environments (including other species) that would yield a path that leads from humans back to the common ancestor with bats, and again forward to modern bats (ignoring some difficulties such as frequency-dependent fitness).

Third, intuitions about getting stuck in local peaks based on the three-dimensional representation as in figure 3 must be treated with care. There are, in fact, a great number of problems with the concept (Provine, 1986, in his biography of Wright, gives a thoughtful critique). First of all, as Wright indicated, an actual genome consists of many (tens of) thousands of genes. Hence, the adaptive landscape has tens of thousands of dimensions, rather than just 3. That makes a big difference, because whereas local peaks seem extremely likely in 3 dimensions, they are in fact increasingly less likely with more and more dimensions. But, perhaps more importantly, the genotype space in Wright's graph is continuous, whereas the genotypes of actual organisms are discrete. Wright's landscapes, as drawn here, can in fact never be constructed for a real example.

Wright and others have looked at other versions of the adaptive landscape that are, in contrast, rigorously defined. One approach is to choose the gene frequencies and population average fitness as axes. A population, in this representation, is then a single point in the landscape. The advantage of this representation is that it ties in nicely with the mathematical model of equation (9). However, the disadvantage is that in such a landscape one cannot visualise the effects of selection, mutation, genetic drift and subdivision of the population, which was the whole point of introducing the metaphor.

Alternatively, one can choose to use phenotypic, continuous traits against individual fitness as the axes of the landscape. The disadvantage of this approach is that mutations, which define what a genotype's "neighbours" are, are of course defined genotypically. Therefore, the random variation that builds up by mutation, will not generally be centred around a single population mean in phenotypic space. In cases where very little is known about the genetics anyway, such as language, that might not really matter, but, as we will see, there the landscape cannot be constructed anyway because of frequency dependence.

Nevertheless, the view of evolution as optimisation yields a powerful approach for deriving predictions about an evolving system, or for understanding an evolved system as adapted for a specific purpose. Parker & Maynard Smith (1990) present a methodology for evolutionary reasoning based on this view which they call "optimality theory"[2]. They first emphasise that every evolutionary study must start with identifying a clear biological question. Step 2 is to identify a set of strategies that are available for evolution to choose from. Step 3 is to identify a pay-off function, which evolution is supposed to optimise, and to show that the observed biological phenomenon tends towards the optimum. Step 4 is to relate pay-off, which is an indirect measure for fitness, to actual fitness. Finally, step 5 is to derive predictions and test them empirically.

This scheme provides a coherent framework for thinking about the evolution of language, and it is essentially the approach I have taken in this paper, although I have and will put some extra emphasis on specific implications of the approach relevant for language evolution. Note however, that the mathematical models discussed so-far concerned changes in gene frequencies, whereas Optimality Theory and language evolution research are concerned with phenotypic traits that typically involve many, often unknown genes. I will first discuss some limitations of the optimality view that apply even when we look at traits controlled by a single gene, and then discuss the more difficult issue of going from single-gene models to the evolution of complex phenotypic, traits such as language.

5 Limits to Optimality

> *"Natural selection tends only to make each organic being as perfect as, or slightly more perfect than, the other inhabitants of the same country with which it comes into competition.*

[2]Parker & Maynard Smith's (1990) Optimality Theory is completely unrelated to Optimality Theory (Prince & Smolensky, 2004) in linguistics.

And we see that this is the standard of perfection attained under nature" (Darwin, 1872, p 163; quoted in Provine 1986, p209).

As Darwin was well aware, the fact that evolution can be understood as optimisation does not imply that the features of organisms are optimal or perfectly adapted to their environment. The most obvious evidence for the existence of limits to optimality, are the many examples of indigenous species that are rapidly driven to extinction after humans introduced a foreign competing species. There is a whole tradition of listing the limitations of natural selection (e.g. Dawkins, 1982; Barton & Partridge, 2000). These can be roughly classified in four classes: (i) biophysical and genetic constraints, (ii) the speed of evolution, (iii) mutational load and (iv) fluctuating fitness.

With regard to **biophysical constraints**, it is clear that all of the complexities of biological organisms need to grow out of a single cell. Throughout its development, an organism needs to maintain its metabolism, to selectively take up chemicals from its environment and to autonomously build-up all of its complex features. That process of biological pattern formation is constrained by what is possible at all with the materials available in a biotic environment, by what can be coded for by genes, and by which possibilities are reachable for evolution. It is obvious that these constraints are at work, given for instance the limitations in speed of both a prey and a predator trying to outrun each other. It is also obvious, however, that these limitations have not prevented evolution from building exquisitely complex and well-adapted organs such as, for instance, the human ear.

Population and molecular genetics make some specific predictions on **genetic constraints**. Natural selection can often not optimise all different phenotypic traits independently from each other, because of the following features of genes:

- A single gene typically has an effect on many different phenotypic traits (pleiotropy);
- The effect of a gene on a trait depends on the presence or absence of other genes (epistasis);
- Genes are physically linked to each other in a chromosome (linkage).

The little that is known about human genetics relevant for language (e.g. Lai *et al.*, 2001) suggests, unsurprisingly, that all these general observations hold for language as well. The general observation have played a role in a debate about whether or not the Baldwin effect – where initial learnt traits are "assimilated" by genetic evolution – is likely to have played a role in the evolution of complex language (Hinton & Nowlan, 1987; Briscoe, 2000; Yamauchi, 2001; Briscoe, 2003) Nevertheless, it seems too little is known about human genetics to inform specific models of the evolution of language, so they will not play a role in this paper.

Most of these biophysical and genetic constraints are reflected in the choice of the strategy set, which contains all strategies/trait values that are available to evolution, and excludes those that cannot be instantiated. The physical linkage between genes, however, is – in the long term – not one of these hard constraints on what can evolve, because recombination will eventually break the linkage such that one gene can occur without the other. Linkage does constrain how fast things can evolve, which is also crucial for the course of evolution.

More generally, the **speed of evolution** is constrained by the available genetic variation at every step (including effects from linkage) and the strength of selection. Considerations about evolutionary time should be included in evolutionary explanations:

Requirement 6 (Sufficient time) *Evolutionary explanations need to establish that there has been enough time for favourable alleles to get established in the population.*

Evolution needs variation to operate on, and mutation is the source of this variation. However, because mutation is indiscriminate and random, it will also constantly create individuals that are worse than average, or even unviable. This is called **mutational load**. In the adaptive landscape metaphor, whereas selection will push a population to the top of an adaptive peak, mutation will pull the population down-hill. The dynamic equilibrium is called *mutation–selection balance*. For specific cases, such as the evolution of RNA molecules, the constraints on optimisation posed by mutational load can be worked out. For the case of language, again too little is known of its genetic basis to derive any specific limitations. However, since a series of formal models of the cultural transmission of language have been proposed (Nowak *et al.*, 2001; Komarova *et al.*, 2001; Mitchener & Nowak, 2002) that are based on the concept of mutational load, it is worth looking in a bit more detail at how this concept has been formalised.

Eigen (1971) and colleagues generalised the Fisher-Wright equations for evolution with mutation and selection at a single locus, to dynamics with an arbitrary number of loci. Using notation loosely based on Maynard Smith & Szathmáry (1995) and Nowak *et al.* (2001), we can write Eigen's equation as follows:

$$\Delta x_i = \sum_{j=1}^{M} (x_j w_j \mathbf{Q}_{ji}) - \overline{w} x_i, \qquad (10)$$

where i and j are indices for all the M distinct possible genotypes. Δx_i stands for the changes of the frequen-

cies of all genotypes i (hence, the expression (10) defines a system of equations, all the same form and one for each possible i). x_i is the frequency of genotype i and w_i its fitness. $\mathbf{Q}_{ji}$ is the probability that a given child will have genotype i if her parent has genotype j. Hence, $\mathbf{Q}$ is an extremely large matrix of size $M \times M$ that describes the effects of mutation. Finally, $\overline{w}$ is the average fitness in the population; the last term ensures that the effects of selection are relative to the population average fitness.

Eigen looked at a very specific choice of parameters. Suppose that there is a single genotype with a high fitness, and all other genotypes have the same, low fitness. That is, the adaptive landscape is flat, except for a single high peak. Now suppose there is a constant probability μ of mutation per gene, and no cross-over. The probability q that an individual (here: an RNA-molecule) when it reproduces produces identical offspring is now:

$$q = (1 - \mu)^l , \tag{11}$$

where l is the genome length. q is called the "copying fidelity". With a bit of algebra one can work out where the mutation–selection balance is for different mutation probabilities, and thus different copying fidelities. Eigen's exciting result is that there is a precise value of q where the mutation–selection balance suddenly drops to vanishingly small quantities of each possible genotype. That is, if the mutation probability is above a threshold value – the *error threshold* – selection ceases to play any role, and individuals have essentially random genotypes:

Requirement 7 (Mutational load) *Evolutionary explanations need to postulate a mutation rate high enough to generate the variation needed, but low enough to not suffer from an extreme mutational load (to cross the error threshold).*

A final category of limits on optimality comes from **fluctuating fitness**, that is, from the fact that the fitness regime of organisms is constantly changing. First of all, there are temporal fluctuations in the environmental conditions on many different timescales, both regular and irregular: from the day and night cycle to climate changes. Similarly, there are geographic differences, such that migrating organisms might find themselves in very different habitats. Organisms adapted to one set of conditions, are not necessarily adapted to other conditions. A language that evolved for communication between hunter-gatherers on the savannah, is not necessarily adaptive in a modern city environment.

But perhaps more interesting is the situation where the fitness regime of a particular species changes due to evolutionary changes of the species itself (**frequency dependent selection**) or of any of the other species it interacts with (**co-evolution**). The evolution of language and communication is frequency-dependent, because linguistic innovations are unlikely to pay off if there is no one to talk to. The fitness coefficients in language evolution are therefore not constants, as in equation (8), but will depend on the frequencies of the different alleles in the population. Evolutionary game theory is the general framework for addressing frequency-dependent selection, and will be discussed in the next section. Because natural languages are transmitted culturally, there can also be a process of cultural evolution, such that we can perhaps sensibly speak about the *coevolution of language and the brain* (Deacon, 1997; similar ideas were explored earlier in e.g. Christiansen, 1994; Kirby, 1994). This is explored a bit further in section 10 in general terms.

A related phenomenon is **sexual selection**, where selection is not on the ability to survive to reproductive age or the ability to reproduce per se, but on the ability to beat rivals of the same sex in the competition for a mate, or on the ability to persuade potential sexual partners to choose one as a mate (Darwin, 1859, p.94). Here, the fitness of a given genotype (defining e.g. a male trait) is not fixed, but also dependent on the frequency of all the possible genotypes (regulating e.g. female preferences) in the population. Exotic, maladaptive traits that are due to sexual selection, such as the ornate peacock-tale or the violent and sometimes lethal *love darts* in hermaphrodite snails, are nice examples of the suboptimal traits that can result from frequency dependent selection. In the evolution of speech, sexual selection seems to have played a role in shaping the secondary sexual traits, such as the lower pitch in human male voices, which results from larger larynx and vocal folds, and a change in formant frequencies at puberty, which makes males appear larger and results from a second descent of the larynx. More controversial are ideas about the role of sexual selection in the evolution of the first descent of the larynx (that happens in both males and females in the first few months after birth, Lieberman, 1984; Hauser & Fitch, 2003), and in the evolution of complex syntax (Pinker & Bloom, 1990).

6 Phenotypic Evolution

We have seen that evolution can be understood as a process of optimisation, but under a range of constraints and with continuously shifting targets. The constraints and trade-offs are all crucial elements of adaptive explanations. In fact, without such constraints, the notion of "adaptation" would be meaningless: without constraints and trade-offs, only almighty beings would exist. The more precise we can be about constraints and trade-offs, including about genetic details, the more convincing demonstrations of optimality within these constraints

are as evolutionary explanations. However, even without a complete understanding of the genetic constraints, we can make progress in understanding evolution at the phenotypic level, by incorporating likely constraints in formal models and deriving testable predictions.

As an example of the structure of such optimality arguments, consider the evolution of hearing and suppose that it can be described with a single variable: the threshold value θ for signal detection. Presumably, the benefit is maximal when this θ approaches 0 (assuming the brain can select and process the information it needs), and the benefit approaches 0 when θ is infinitely large. The cost of an infinitely small θ is infinitely big, however, because biophysical constraints dictate that infinitely small θ requires infinitely large ears. With very large θ we could do away with ears all together and have a cost approaching 0. When we subtract the cost from the benefit, we get the payoff function. If the cost and benefit function adequately describe the selection pressures and constraints, we expect the evolutionary dynamics to lead to the optimum of the payoff function, shown qualitatively in figure 4. Now, if we could find a combination of benefit and cost functions, and empirical observations of θ in nature that match the predicted optimum, that would constitute strong evidence for either the hypothesis that θ evolved for the function described by the payoff function, or – if we are already confident of the adaptive function – that the hypothesised constraints, described by the cost function, were the right ones.

Can we make a similar analysis of the evolution of key features of natural language? That is, can we identify the payoff function and its optimum under relevant constraints and show that natural language corresponds to that optimum? Unfortunately, we know relatively little about the biophysical and genetic constraints, the relevant mutations in the evolution of language and the neural implementation of our linguistic abilities. It is therefore difficult to make precise what strategy set was available for evolution. The best examples of trade-offs in language are probably in the physical properties of speech. Liljencrants and Lindblom's (1972) demonstration that the vowel systems in human language appear to be optimised for reliable recognition under noisy conditions and under constraints on perception and articulation, is suggestive. Lieberman (1984) has argued that the human larynx has descended deeper down the throat in order to allow more flexibility of the articulatory organs. This allows us to make many different speech sounds, at the expense of an increased propensity to choke. Although controversial (Hauser & Fitch, 2003), this theory on the evolution of language does illustrate the role of evolutionary trade-offs that result from the physiological constraints in speech production.

For other components of human language, such as its semantics or syntax, it is extremely difficult to derive biophysical constraints. What sort of grammars can or cannot be encoded by genes and implemented in neuronal tissue? The only solid results relevant to this question, suggest that quite a variety of networks of interacting cells are *Turing equivalent*. That is, they can – if sufficiently large, given sufficient time and properly initialised and interpreted – compute any computable function (Siegelmann & Sontag, 1991; Wolfram, 2002). This is not to say that any grammar can be easily encoded by genes or acquired by a neural net; but without better models of the neural implementation of language, we cannot start to make sensible assumptions about the actual architectural constraints on natural language syntax that were at work during human evolution. This is how I interpret Chomsky's well-known reservations about the feasibility of scientific explanations of the evolution of language, such as expressed in this famous quote:

> *"We know very little about what happens when 10^{10} neurons are crammed into something the size of a basketball, with further conditions imposed by the specific manner in which this system developed over time. It would be a serious error to suppose that all properties, or the interesting properties of the structures that evolved, can be 'explained' in terms of natural selection." (Chomsky, 1975, p.59).*

However, it would be overly pessimistic to conclude – as Chomsky seems to do – that we can therefore not say anything sensible about how language evolved. There are two categories of constraints in language evolution that can be made precise. First of all, we have good "mentalist" models of syntax that describe its fundamental computational properties, and the **computational constraints** that any implementation will face. For instance, we know there exist constructions in natural languages that cannot be modelled by weaker formalisms (in terms of the extended Chomsky Hierarchy) than (mildly) context-sensitive rewriting grammars (Joshi *et al.*, 1991); we know that the whole class of context-sensitive rewriting grammars is not *identifiable in the limit* from positive samples alone (Gold, 1967); and we know that grammars of that type have a worst-case time-complexity of $O(n^5)$ in parsing (Barton & Berwick, 1987). Such computational constraints on representation, learning and processing, and the formalisms they are expressed in, allow us to at least make a start with testing the internal consistency of an evolutionary scenario, and with formulating a sensible strategy set for evolution.

Second, there are constraints that follow from the **social, communicative function** of language. Humans use natural language to communicate with others, on the average for many hours a day per person. This re-

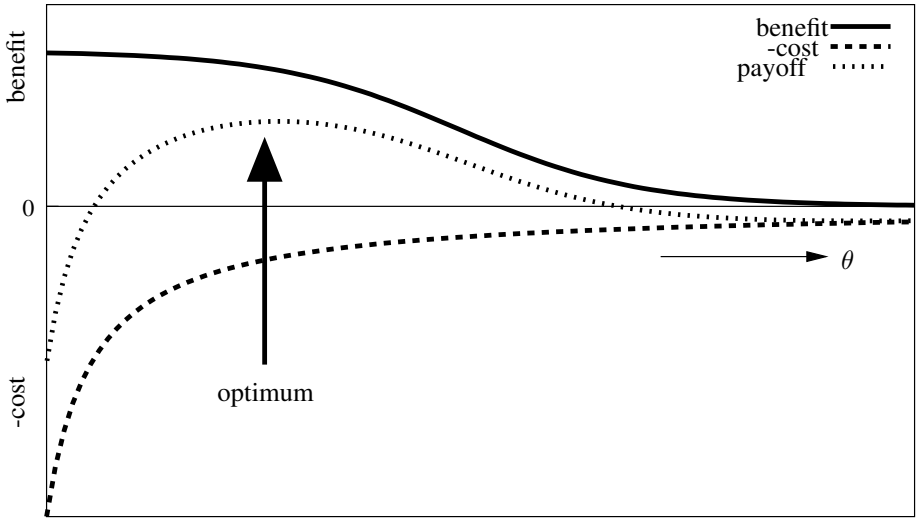


Figure 4: Evolutionary optimisation under biophysical constraints. The graph sketches the benefits (top curve) and costs (bottom curve) for a continuous range of detection thresholds θ (x-axis) in the evolution of hearing. An extremely low threshold (left end) is very useful, but also very costly; an extremely high threshold (right end) is very cheap, but not of much use. The optimum of the payoff function (middle curve) is therefore at an intermediate value of θ.

quires a shared code, such that both speakers and hearers understand the meanings of utterances. Moreover, it requires the willingness of the speaker to give away information and, at least in general, to be truthful, as well as a willingness from the hearer to listen and interpret the message received. These issues can be addressed in the framework of evolutionary game theory, which will be discussed next.

7 Evolutionary Game Theory

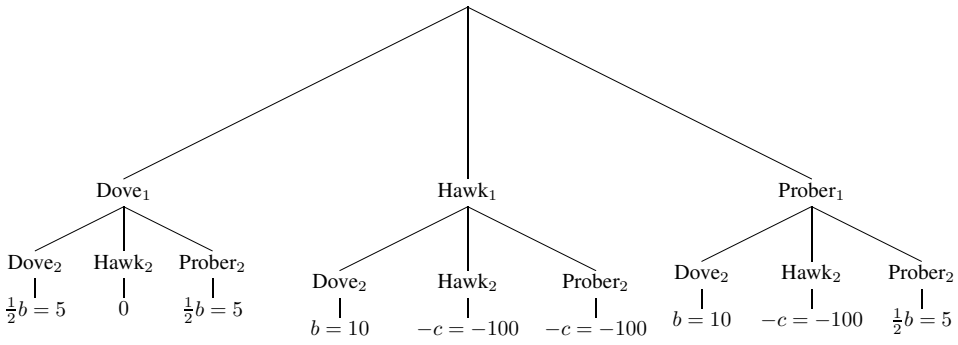
The evolutionary history of human language can be viewed as a process of phenotypic optimisation, under (largely unknown) biophysical and cognitive constraints that determined which communication systems were possible at all, and in a social–communicative context that determined which systems were better than others, but that continuously shifted the evolutionary targets because the frequency of a linguistic trait in the population influences its usefulness.

The formal framework to describe the consequences of multiple agents optimising their own payoff in a social context is the **Theory of Games**. Game theory conceptualises the interaction between agents, the "players", as a game where all players choose from a set of available strategies. Crucially, the outcome of a game for each player, its payoff, depends on the strategies of other players. Unlike the example in figure 4, where payoff is a function of the player's own strategy alone (the trait value, θ), in game theory the payoff is a function of both the player's strategy and the strategies played by other players.

The following example is derived from Maynard Smith & Price (1973). Imagine a conflict between two birds competing for a single food source, each with the choice between three strategies: "dove" (retreat immediately if the other player is aggressive), "hawk" (always be aggressive) and "prober" (start off aggressive, but share the food source peacefully if the other player does not give up, but does not escalate either, and continue aggressively if the other player does give up). If the value of the food source is $b = 10$, and the expected cost of an escalated fight $c = 100$, the possible payoffs for player 1, given her and player 2's decisions, are given in figure 5(a). For 2 players and a small number of discrete strategies, this can be conveniently summarised with a *payoff matrix*, as in figure 5(b).

We can postulate a decision mechanism for each player, and study how the outcome of the game changes with players adapting their strategies based on what the other players do. The dynamics of such games, with all players making their own decisions, are often extremely difficult to describe. Often, however, it is possible to derive the conditions under which a game is stable. In non-cooperative game-theory – where "selfish" players each try to optimise their own payoff – the crucial concept is that of a **Nash equilibrium** (Nash, 1950)[3]. This equilibrium is defined as the situation where no player can increase her payoff by unilaterally changing her strategy. Thus, for any n-tuple of pure strategies (one for each player) the Nash equilibrium requires that each player's strategy maximises her expected payoff against all other $n - 1$ strategies.

[3]Grafen (2003) attributes the discovery of the Nash equilibrium to William Waldegrave, 1713, and refers to A. Hald (1990), "A History of Probability and Statistics and Their Applications before 1750", New York: Wiley Interscience.



(a) extensive representation

	player 2's strategy		
player 1's strategy ↓	Dove	Hawk	Prober
Dove	5	0	5
Hawk	10	-100	-100
Prober	10	-100	5

(b) payoff matrix player 1

Figure 5: Extensive and matrix representations of games

The Nash Equilibrium plays a major role in modern economic theory, as *rational* players are assumed to maximise their payoff, and games will therefore typically evolve toward a Nash equilibrium. Other branches of economic game theory make different assumptions on what is optimised, and sometimes use different stability concepts. For instance, cooperative game-theory – where players are assumed to try to optimise the average payoff of all players in the game – uses the concept of "Pareto optimum", where no player can increase her payoff without decreasing the payoff of another player. In the theory of bounded rationality (Simon, 1955, 1969), the consequences of limitations in knowledge are investigated, where players are not necessarily maximising, but rather *satisficing* their payoffs.

In evolutionary biology (after some pioneering work by R.C. Lewontin and W.D. Hamilton, as is discussed in Maynard Smith, 1982) the use of game theory took off with the work of Maynard Smith & Price (1973) and Maynard Smith (1982). Maynard Smith & Price introduced the concept of **Evolutionarily Stable Strategy** (ESS) in an analysis of the evolutionary advantages of "limited war" strategies in animal conflicts, such as the prober strategy introduced above. An ESS is a strategy that cannot be *invaded* by any other strategy, because all other strategies get either a lower payoff when playing against the ESS, or if their payoff is equal, they get a lower payoff when playing against themselves. That is, if $F(i, j)$ gives the payoff for a player playing strategy i against an opponent playing strategy j, then i is an ESS if for every strategy j either $F(i, i) > F(i, j)$ or $F(i, i) = F(i, j) > F(j, j)$. Every ESS also defines a Nash Equilibrium, but the stability criterion is stricter, because it implies that every alternative strategy will be selected against if it occurs at small but non-zero frequency in the population.

In the example of figure 5, we can see that the dove-strategy is not an ESS, because the hawk-strategy has a higher payoff when playing against it. In a populations of doves, the hawk strategy thus enjoys an initial selective advantage and will increase in frequency. The hawk-strategy is not an ESS either. A population consisting of just hawks can in turn be invaded by the dove-strategy, which has a higher payoff in a population of hawks, or by the prober-strategy, which has equal payoff against hawk but a higher payoff against itself. Only the prober strategy, in the present simple model, is an ESS: both doves and hawks fare worse than the prober in a population of probers[4].

If we exclude the prober-strategy from the strategy set, the resulting hawk-dove game has no ESS, i.e. a population of individuals all playing one pure strategy, can be invaded by the other strategy. In such games there might still be a stable distribution of phenotype frequencies in a population – called an **Evolutionarily Stable State**. In such a situation, there are distinct,

[4]In the original paper (Maynard Smith & Price, 1973), this game was introduced with "dove" labeled "mouse" and "prober" labeled "prober-retaliator". Incidentally, an unfortunate choice of parameters resulted in there being in fact no ESS at all, even though a fourth strategy "retaliator" was erroneously identified as such.

genetically different players in the population ("polymorphism"), and this polymorphism is maintained by selection. Interestingly, such a stable distribution with p doves and $1-p$ hawks is equivalent to a population where each individual plays the dove-strategy with *probability* p and the hawk-strategy with probability $1-p$. If such *mixed strategies* are included in the strategy set (that is, allowed according to the hypothesised constraints), it is an ESS[5] and there is no polymorphism maintained.

The techniques and formalisms from evolutionary game theory immediately lead to some fundamental observations on the evolution of communication. Consider the evolution of an alarm call system similar to the calls that, for instance, ground squirrels (Sherman, 1977) or vervet-monkeys (Seyfarth *et al.*, 1980) use to inform conspecifics of the presence of predators. If we focus on just two signals, 1 and 2, and just two types of predators, aerial (E, e.g. eagles) and terrestrial predators (L, e.g. leopards), we can postulate the following strategy set:

Sender strategies	A:	send 1 when observing E; send 2 when observing L.
	B:	send 2 when observing E; send 1 when observing L.
	C:	never send anything.
Receiver strategies	A':	act as if observing E when hearing 1; act as if observing L when hearing 2.
	B':	act as if observing E when hearing 2; act as if observing L when hearing 1.
	C':	ignore all received calls.

In the case of alarm calls, the payoffs for sender and receiver are very different. The sender will suffer a cost, because by calling she alerts the predator of her presence and location. Evidence of the existence of a real cost in nature comes from the fact that alarm calls typically have very high pitch, which makes it more difficult for predators to locate the caller (Maynard Smith, 1982). The payoff matrix for the sender will therefore have all negative entries (parameter c) for strategies A and B, and (by definition) 0 for strategy C.

The receiver, on the other hand, will profit from a call *if and only if she correctly interprets it*. That benefit is quantified with parameter b. If the actual predator is a leopard, acting as if an eagle is observed can be a costly mistake: monkeys flee into the bushes to escape from an eagle attack, but that is in fact exactly where leopards hide (Seyfarth & Cheney, 1997). The cost of mis-interpretation is quantified as parameter m. If the receiver ignores all calls, her payoff is 0 (again, by definition). The payoff matrices in this simple example will thus look as in figure 6.

sender strategy ↓	receiver strategy A'	B'	C'
A	$-c$	$-c$	$-c$
B	$-c$	$-c$	$-c$
C	0	0	0

(a) sender's payoff

sender strategy ↓	receiver strategy A'	B'	C'
A	$+b$	$-m$	0
B	$-m$	$+b$	0
C	0	0	0

(b) receiver's payoff

Figure 6: Payoff matrices in a simple alarm call system

It is clear that neither A nor B can be the stable strategy for the speaker; if the cost of calling, c, is non-negligible, the strategy of not communicating at all, C, is always optimal. In explaining the evolution of communication, we thus face a **problem of cooperation**: if the benefits of communication are for the hearer, the sender has no incentive to give away her information, or even put herself at risk. Dawkins & Krebs (1978) pointed out this problem with what they call the "classical ethological" view on animal communication, which takes communication as existing for the benefit of the group. Dawkins and Krebs have therefore suggested that communication should be understood as a form of manipulation, with the benefits of successful manipulation with the sender.

Others (e.g. Maynard Smith, 1965; Sherman, 1977; Cavalli-Sforza & Feldman, 1983) have argued that "altruistic" communication can evolve through kin selection. However, the appropriateness of kin selection for human language – where communication is typically with non-kin – has been called into question (Dessalles, 1998). Dessalles has instead argued for a form of "reciprocal altruism", where there is a real benefit for the sender, because it is rewarded with status in the population. Fitch (2004) reviews his and other arguments, but concludes that they are not convincing. He posits the "mother tongue" hypothesis – that human language developed primarily in a context of kin communication – as one of a number of factors that shaped human language in its evolution, and calls for further exploration of the role of kin selection in language evolution.

In many circumstances, for instance sexual signaling, the problem is not so much in the willingness to send

[5] Grafen (1979) points out that mixed strategy ESS's and pure strategy evolutionary stable states are not equivalent in kin selection models.

signals, because the senders benefit, but in the **honesty** of the signals. A large amount of work on the evolution of animal and human communication has been concerned with this problem, leading to what is now called "honest signaling theory" (the handicap principle, Zahavi, 1975, 1977; Grafen, 1990). Hence, the problem of cooperation is pervasive in work on the evolution of communication, although its instantiations differ with different assumptions on the costs and benefits of communication, for both sender and receiver. Although the problem of cooperation is a consequence of careful considerations of payoff, strategy sets and invasibility, I will, because of its importance, add it as a separate point to the list of requirements of evolutionary explanations:

Requirement 8 (Problem of cooperation)
Evolutionary explanations of the evolution of language need to address the problem of cooperation, and demonstrate that senders will be willing to send honest signals, and that hearers will be willing to receive and believe the signal.

Even if we find a scenario where successful communication is in the interest of both the speaker and the hearer, there is another problem that arises from the frequency-dependence of language evolution. We could call this the **problem of coordination**. If we ignore the non-cooperative strategies C and C', how does a population of players coordinate their behaviours such that they play either A and A', or B and B'? That is, how do they agree on a shared code? This problem seems particularly difficult when we consider a series of innovations, as in Jackendoff's (2002) scenario of the evolution of human language. Each of these innovations needs to confer a fitness advantage if it is to spread the population, but it is difficult to see how a genuine innovation can be advantageous to the individual if it is not shared by the rest of the population (Zuidema & Hogeweg, 2000; Zuidema & de Boer, 2003).

Lewis (1969) showed that only "perfect" communication systems are "separating equilibria", which, if the role of "rationality" of the players is replaced by natural selection, corresponds to evolutionary stable states (Skyrms, 1996; Trapa & Nowak, 2000; van Rooij, 2004). Models in this tradition make the following assumptions:

- There is no cost to communication;
- The interests of sender and receiver are perfectly aligned;
- There is a discrete set of signals and a discrete set of meanings, and the number of signals equals the number of meanings;
- All meanings are equally frequent and valuable;
- Every "perfect" mapping from meanings to signals is equally good (which implies that meanings have no relation to each other, signals have no relation to each other, and meanings have no natural relation to signals);
- The meaning–signal associations are innate and inherited from parent to child.

It is easy to see why perfect communication systems are the only ESS's under these assumptions: if a communication system is sub-optimal, there must be synonymy: multiple signals are used for the same meaning. For the sender, however, it is always best to express a meaning m with the single signal s that has the highest chance of being understood, i.e. to avoid synonymy. The alternative signal(s) will thus not be used to express m anymore, and becomes available (through drift) for meanings that cannot be expressed yet. Hence, only "perfect" systems are stable against selection and drift.

It is clear, however, that all of these assumptions are violated in reality. Signals do have a cost, interests are not perfectly aligned, meanings and signals are not discrete, symbolic entities, but have similarity relations with themselves and each other, and, at least in human language, meaning–signal mappings are learnt and not innate. The problem of coordination thus remains a major open issue in the evolution of language, which we can add to the list of requirements:

Requirement 9 (Problem of coordination)
Explanations for the evolution of language need to deal with the problem of coordination, that is, show how, after each innovation, a shared code can be established and maintained.

Much of the work on the evolution of language can be seen as dealing with this problem. A number of models, for instance, relax the innateness assumption above, and study, in computer simulations, the evolutionary success of a number of different strategies in word learning (Hurford, 1989; Oliphant, 1999; Smith, 2004). The payoff function in Hurford's model is the expected success in communication between a sender and a receiver (i.e. the game is cooperative; both sender and receiver benefit from success). Sender behaviour is characterised by a probabilistic mapping from a set of M meanings to a set of F signals; receiver behaviour by a probabilistic mapping from the signals to the meanings.

Hurford was interested in how these functions were learnt, and in the evolution of different learning strategies. The strategy set Hurford considered consisted of three strategies, termed imitator (that imitates the observed average sending and receiving behaviour in the population), calculator (that estimates the best send and

receive functions based on observations of the population's receive and send behaviour respectively) and Saussurean learner (that chooses the same receive function as the calculator, but derives the send function from that receive function rather than from the receiving behaviour in the population). Hurford showed that Saussurean learners outcompete the other two learning strategies. These results were extended by Oliphant & Batali (1996), Oliphant (1999) and Smith (2004), among others. From these studies it emerged that learning strategies can evolve that give rise to "perfect" communication systems in a population.

Other models (e.g. Nowak & Krakauer, 1999), do not model such explicit learning rules, but do relax some of the other assumptions mentioned. More work is needed to study whether the results from these studies hold when learning is modelled explicitly. An encouraging result in this respect is due to Calvin Harley (1981). He studies the evolution of learning rules and showed that evolution will favour rules that *learn* the evolutionary stable strategy. Hence, results on Evolutionary Stable Strategies in innate communication systems, in principle carry over to situations where the same strategies are acquired in a learning process (Maynard Smith, 1982, chapter 4).

8 Levels of Selection

I have discussed some basic concepts from population genetics, which describes the change in frequencies of *genes*, and from evolutionary game theory, which describes the invasion and replacement of phenotypic *strategies* of individuals. The two approaches are obviously related, because the fitnesses of genes are dependent on the phenotypes they code for, and a strategy will only replace another strategy if all the genes necessary for that strategy are selected for and get established in a population. But the description of the evolutionary process in population genetics and evolutionary game theory are set at entirely different levels.

In Dawkins' (Dawkins, 1976) terminology, genes are *replicators*: they are the bits of information that get copied and transmitted – more or less intact – to the next generation. Individuals are *vehicles* (Dawkins, 1976) or *reproducers* (Szathmáry, 1999). In sexual species, such as humans, a child is radically different from any one parent, because she inherits only 50% of the genes. Individuals, therefore, are not replicators, even though they are the obvious level of description when we talk about fitnesses and strategies.

If *replicators* and *reproducers* were the same objects, evolutionary dynamics would be relatively easy to describe. But in general, especially in sexual species, they are not. Genes are "packaged" – contained within the structured genome of an individual that lives within a structured population. That packaging makes the fate of a specific gene depend on the other genes it is associated with (genes that occur together more often or less often than would be expected on the basis of their frequencies alone, are said to be in *linkage disequilibrium*). If a gene a happens to be associated with a gene b that is under strong positive selection, gene a will increase in frequency even though it does not itself contribute to the fitness of its carrier ("genetic hitch-hiking", Hill & Robertson, 1966; Maynard Smith & Haigh, 1974). To predict the fate of a specific gene, we therefore need to know the statistical associations with other genes.

To make things even more complicated, not just the gene frequencies change; also the associations themselves change in evolution. The *physical linkage* between genes on a chromosome tends to keep these genes together, but *recombination* breaks up these associations; *sexual selection* generates associations between for instance, the preferences of the females and the selected traits of the males; finally, *epistasis* also generates linkage equilibrium, because if genes are much better in combination than they are apart, natural selection itself will make the combination more frequent than expected by chance. Barton & Turelli (1991) and Kirkpatrick, Johnson & Barton (2002) have developed a mathematical framework to describe the dynamics of such *multi-locus evolution*; however, they take fitnesses as given and do not yet provide a bridge to the fitness concept in phenotypic models.

Hence, the relation between gene frequency change and adaptation at the level of the individual (such as language) is not at all trivial. The problem with the gene as the level of description is that we don't know the relevant fitness coefficients, because our knowledge of life, death and reproduction is almost entirely specified at the level of the individual. But the problem with the individual as level of description, is that we are not necessarily justified in assuming that natural selection corresponds to optimisation. Do the results from game-theoretic analyses translate to fitness coefficients of the genes that underlie the strategies? How do we relate the fitness coefficients, and the fundamental results about evolution as optimisation by Fisher and Wright, to adaptation on the level of individuals? Grafen (2003), in a discussion of Fisher's "fundamental theorem of natural selection" (Fisher, 1930) observes that (too) few researchers in evolution worry about these issues:

> *"the theorem was fundamental in 1930 because it isolated the adaptive engine in evolution and made an extraordinary link between gene frequencies and adaptive change. It really did show how Darwinian natural selection worked simply and consistently and persistently amid the maelstrom of complexities of population genetics. The theorem is just as*

important today for that reason. This is not popularly realised by biologists because most take for granted an informal sense that natural selection leads to organisms maximizing their fitness, but they do not ask how that sense can be justified." (Grafen, 2003, p.327)

Grafen lists three assumptions that are made in the original version of Fisher's theorem, and apply equally to Wright's equations discussed in section 4:

- It assumes the fitnesses of genes are frequency independent. That is, the fitness of a given genotype is not dependent on which other genotypes are present and at which frequencies in the population. Consequences of frequency dependence are studied in evolutionary game-theory (Maynard Smith & Price, 1973; Maynard Smith, 1982).
- It assumes that all individuals interact with all other individuals with equal probability. That is, it assumes the fitness of a given genotype is not dependent on the genotypes which are potentially correlated with it. Consequences of such correlations are studied in social evolution theory (Hamilton, 1964a,b; Frank, 1998).
- It assumes fitnesses are fixed; Grafen himself has worked on the consequences of natural selection under uncertainty.

For the purposes of this paper, it would take too far to investigate the contributions of Grafen and others to relate population genetics and evolutionary game theory. However, a few important implications for language evolution research from the discussion so-far are worth making explicit. First, a "strategy" in a game-theoretic analysis will typically be coded for by many genes (*pleiotropy*). So if alleles $a_1, a_2 \dots a_n$ at loci 1 to n are needed for an evolutionarily stable strategy A, we need each of these alleles to represent a step in the right direction. In technical terms, we need *additive genetic variance*; Maynard Smith (1982) argues that additive genetic variance is common in nature, and that this is therefore a reasonable assumption to make in game-theoretic analyses. We need to be aware, however, that we ignore all the phenomena of multi-locus evolution in game-theoretic analyses of language.

Requirement 10 (Levels of selection) *Explanations for the evolution of language need to relate selection at the level of individuals or groups to changes in gene frequencies. That is, they need to specify and relate the assumed levels of description for selection and heritability.*

Second, an important (methodological) observation is that there is no single best level of description; researchers make a heuristic choice about the level at which they will describe the evolutionary dynamics. Every model will only be an approximation, and it depends on the phenomenon of interest at which level the evolutionary process is most adequately described. Below, I will briefly discuss kin selection, and show, using the Price equation, why for the phenomena of social evolution the population structure is a crucial level of description that is left out in standard game-theoretic models.

9 Social Evolution & Kin Selection

The techniques from social evolution theory could fill a whole separate paper; I will therefore keep the discussion brief. One fundamental equation, the **Price equation** (Price, 1970), is useful, however, to highlight a silent assumption in game-theoretic models, and to illustrate the issue of multiple levels of selection. The Price equation is easily derived; I will follow here Frank (1998) and Andy Gardner (p.c.). Like Wright's equation (9), it can be interpreted as describing the change in the frequency of a gene, but more generally it describes the change in the value of any trait z.

Price introduces his equation as follows:

> *"Gene frequency change is the basic event in biological evolution. The following equation [...], which gives frequency change under selection from one generation to the next for a single gene or for any linear function of any number of genes at any number of loci, holds for any sort of dominance or epistasis, for sexual or asexual reproduction, for random or nonrandom mating, for diploid, haploid or polyploid species, and even for imaginary species with more than two sexes" (Price, 1970, p.520)*

We are interested in the change in frequency of a specific trait z in the population between the present ($\overline{z}$) and the next generation ($\overline{z}'$). If we divide up the population in M units $q_1 \dots q_M$ (these units are, for instance, individuals or groups, depending on the level of selection the equation is meant to describe), and we know their fitnesses $w_1 \dots w_M$ and trait values $z_1 \dots z_M$, then the change of the trait's frequency in the whole population is given by:

$$\begin{aligned} \Delta\overline{z} &= \overline{z}' - \overline{z} \\ &= \sum_i q_i' z_i' - \overline{z} \\ &= \sum_i q_i \frac{w_i}{\overline{w}} (z_i + \Delta z_i) - \overline{z} \end{aligned} \tag{12}$$

Multiplying both sides of this equation with $\overline{w}$, and

rearranging gives:

$$\begin{aligned}\overline{w}\Delta\overline{z} &= \sum_i q_i w_i z_i + \sum_i q_i w_i \Delta z_i - \overline{w}\,\overline{z} \\ &= \underbrace{\sum_i q_i w_i z_i - \overline{w}\,\overline{z}}_{\text{Cov}[w,z]} + \underbrace{\sum_i q_i w_i \Delta z_i}_{E[w\Delta z]} \quad (13)\end{aligned}$$

As indicated, the terms in equation (13) correspond, by definition, to the *covariance* between fitness and trait value, and *expected value*[6]. Hence, the process of evolution can be elegantly summarised in the Price equation, as follows:

$$\overline{w}\Delta\overline{z} = \underbrace{\text{Cov}[w,z]}_{\text{selection}} + \underbrace{E[w\Delta z]}_{\text{transmission}} \quad (14)$$

The Price equation partitions the process of evolution into a term that describes the effects of selection (traits that are associated strongly with fitness will be selected for most effectively), and a term that describes the effects of (biased) transmission (the index i is the index of the parent; hence Δz_i describes the change in the trait value – from a particular parent to all its offspring – regardless of selection).

Observe that the transmission term in the Price equation looks very similar to the left-hand side of that equation. This fact allows us to relate different levels of selection. As an illustration, I will here derive Hamilton's (Hamilton, 1964a,b) famous result on kin selection, which says that an altruistic trait can evolve if the benefit b times the relatedness r is larger than the cost c:

$$br > c. \quad (15)$$

The derivation using the Price equation highlights the correct interpretation of *relatedness* and suggests applications for language evolution. The derivation concerns the evolution of an altruistic trait, such as the alarm calls discussed in the previous section. For simplicity, assume an individual either does or does not have this trait. We indicate this with the variable z, that is, $z = 1$ or $z = 0$. We can ask: under which circumstances will this trait evolve?

Consider a population, subdivided (at random) in N groups $G_1 \ldots G_N$, each of size M individuals. In each group G_i, individuals benefit from the amount of altruism in that group, labelled as z_i; the total benefit is bz_i. The jth individual in that group, however, also suffers a cost from being altruistic, indicated with c; the cost is thus cz_{ij}. The fitness of the jth individual in the ith group is now given by:

$$w_{ij} = \alpha + bz_i - cz_{ij}, \quad (16)$$

where α is a baseline fitness (not dependent on the presence or absence of the altruistic trait). The fitness of the ith group is given by:

$$w_i = \alpha + (b - c)z_i. \quad (17)$$

Hence, an individual's fitness (her relative contribution to the total offspring of the group) depends on the amount of altruism received and the amount of altruism given. Obviously, if the cost c of being altruistic is larger than 0, it is always best for an individual to be selfish. The group's fitness[7] (the relative contribution of this group's offspring in the total offspring of the whole population) depends on the total amount of altruism given. If the cost c of altruism is lower than the benefit b, it is always best *for the group* if all individuals are altruistic.

The evolutionary process within each group i can be described with a Price equation, as in equation (14). If we assume there is no transmission bias, the equation simplifies to:

$$\overline{w_{ij}}\Delta\overline{z_{ij}} = w_i\Delta z_i = \text{Cov}_j[w_{ij}, z_{ij}]. \quad (18)$$

The evolutionary process at the level of the whole population is also described with a Price equation, where the transmission term concerns the within-group dynamics of equation (18):

$$\begin{aligned}\overline{w_i}\Delta\overline{z_i} &= \text{Cov}_i[w_i, z_i] + E_i[w_i\Delta z_i] \\ &= \text{Cov}_i[w_i, z_i] + E_i[\text{Cov}_j[w_{ij}, z_{ij}]] \quad (19)\end{aligned}$$

The covariance in above equation can be replaced by a regression and variance term, because (by definition) $\text{Cov}(x, y) = \beta(x, y)\text{Var}(y)$. This gives the following equation:

$$\overline{w_i}\Delta\overline{z_i} = \beta(w_i, z_i)\text{Var}_i[z_i] + E_i[\beta(w_{ij}, z_{ij})\text{Var}_j[z_{ij}]]. \quad (20)$$

These regression terms β can be read off directly from equations (16) and (17), because they correspond to the slope of the fitness functions, i.e. $\beta(w_i, z_i) = b - c$ and $\beta(w_{ij}, z_{ij}) = -c$. Substituting these values into equation (20) and rearranging gives:

$$\begin{aligned}\overline{w_i}\Delta\overline{z_i} &= (b - c)\text{Var}_i[z_i] + E_i[-c\text{Var}_j[z_{ij}]] \\ &= (b - c)\text{Var}_i[z_i] - cE_i[\text{Var}_j[z_{ij}]] \\ &= b\text{Var}_i[z_i] - c(\text{Var}_i[z_i] + E_i[\text{Var}_j[z_{ij}]]) \\ &= b\text{Var}_i[z_i] - c\text{Var}_{\text{total}} \\ &= \left(b\frac{\text{Var}_i[z_i]}{\text{Var}_{\text{total}}} - c\right)\text{Var}_{\text{total}}, \quad (21)\end{aligned}$$

[6] The covariance between two variables x and y is defined as $\text{Cov}(x_i, y_i) = \frac{1}{N}\sum_{i=1}^{N}((x_i - \overline{x})(y_i - \overline{y})) = \overline{xy} - \overline{x}\,\overline{y}$, i.e. the product of the means minus the mean of the products. Expected value of a variable x is defined as $E(x) = \sum_{i=1}^{N} P(x = x_i)x_i$, i.e. the sum of all possible values weighted by the probability of each value. Covariance is the most obvious way of measuring a departure from statistical independence. If x and y vary independently from each other, then $E(xy) = E(x)E(y)$, and the covariance is 0.

[7] Note that, although parent groups are of fixed size M, some groups produce more offspring than others.

where $\text{Var}_{\text{total}}$ is the total variance. This establishes a derivation of Hamilton's rule from the Price equation, because the average relatedness between two individuals in a population, equals the between group variance as a proportion of the total variance. That is, $r = \frac{\text{Var}_i[z_i]}{\text{Var}_{\text{total}}}$. If the benefits of trait z, weighted with the relatedness within a group, are larger than the costs, i.e. $rb > c$, then $\Delta\overline{z}$ will be positive, i.e. evolution will favour the trait even if it harms the individual.

It is important to note that Hamilton's rule is widely misinterpreted. As this derivation shows, the relatedness term r is *not* the fraction of genes two individuals share (*identity by descent*), as is commonly assumed (e.g. Okasha, 2003). Rather, it is a statistical association between the trait of interest in one individual and the trait in the individual she interacts with. Therefore, the relatedness between two individuals can even be negative. This simply means that the individuals are less related to each other than to a random third individual in the population (Hamilton, 1970). If the association is high enough, altruistic traits can be favoured by natural selection[8]. That is, if (for whatever reason) altruists are surrounded by other altruists, they benefit more from the altruism received than from the altruism offered (and conversely, if it is low enough, natural selection can favour *spite* – behaviours that harm both the actor and the recipient; Hamilton, 1970; Gardner & West, 2004).

Interactions within kin-groups (and kin recognition) are an important mechanism for this association to arise (hence the Maynard Smith's term "kin selection"), but not the only one. Subdivision of a population in groups is another mechanism (such "group selection" is thus a form of kin selection). Hamilton himself suggested a third mechanism, that of "green beards". If the same gene complex that codes for an altruistic trait, also codes for an external marker (i.e. a green beard), altruists can choose to interact preferentially with each other. This is of interest for language evolution, because language itself could be such a green beard, if individuals with a linguistic innovation can recognise each other based on features in their language. Finally, reciprocal altruism (Trivers, 1971), where players remember the interaction history with other players and play altruistically only against players that have been altruistic in the past, can be understood in the same framework.

Kin selection seems the most promising solution for the problem of cooperation that I introduced in section 7. It would certainly be worthwhile to study formal models of kin selection, that take into account the details of human communication. In this paper, however, I will no further address kin selection or the problem of cooperation. Instead, I will assume the willingness to cooperate exists in modeled populations, and focus on the problem of coordination.

10 Cultural Evolution

Dawkins (1976) emphasised that the principle of natural selection is not restricted to genes or individuals (as Fisher, Wright, Haldane, Price, Hamilton and others were well aware). In every situation where one can identify replicators, heritable variation and natural selection, a process of adaptation can take place. For instance, cultural inventions (or "memes", Dawkins, 1976) – religion, technology, fashion or indeed words and grammatical rules – undergo evolution if there are mechanisms for cultural transmission and cultural selection.

Since Dawkin's book, many wildly speculative theories have been launched under the heading "memetics", which have given this new field a bad reputation. Nevertheless, the basic idea is sound and open to serious investigation (Mesoudi *et al.*, 2004). For a start, all mathematical models and requirements discussed in this paper apply, mutatis mutandis, to cultural evolution as well. The idea of viewing historical language change as a form of evolution is particularly attractive because, on the one hand, it makes the extensive mathematical toolkit of evolutionary biology available to linguistics, and on the other hand, is presents evolutionists with an enormous body of new data.

We need formal models of the cultural evolution of language, in which we can deal with all the constraints on evolutionary models that I listed in this paper. Although many authors have noted the parallels between biological evolution and language change, including Darwin (1871, p.91), only recently have people started to study the cultural evolution of language in such a formal framework. Some relevant mathematical models are those of Cavalli-Sforza & Feldman (1981), Niyogi (2002) and Yang (2000). These authors look at the competition between two or more languages, with no qualitative differences between languages. Simulation models such as those of Kirby (1998) and Batali (2002) look at more open-ended systems, with more explicit formalisms for grammar and learning.

One problem is that is not so easy to decide on the ap-

[8]Darwin already understood the essence of kin selection when he wrote: ""[...] selection may be applied to the family, as well as to the individual, and may thus gain the desired end. Thus, a well-flavoured vegetable is cooked, and the individual is destroyed; but the horticulturist sows seeds of the same stock, and confidently expects to get nearly the same variety; breeders of cattle wish the flesh and fat to be well marbled together; the animal has been slaughtered, but the breeder goes with confidence to the same family. [...] Thus I believe it has been with social insects: a slight modification of structure, or instinct, correlated with the sterile condition of certain members of the community, has been advantageous to the community: consequently the fertile males and females of the community flourished, and transmitted to their fertile offspring a tendency to produce sterile members having the same modification." (Darwin, 1859, p.258-259)

propriate units of selection. For instance, Kirby (2000) described the dynamics in his simulation model with context-free grammar rules as replicators under selection for more reliable replication. In later papers, however, he argued that the analogy between biological and cultural evolution in this case breaks down (Kirby, 2002). This is because the grammatical rules are *induced* from observable language, whereas in biological evolution genes are *inherited*, with no feedback from phenotype to genotype (other than through the effects of selection). This is known as the "central dogma of molecular biology". This observation is correct, of course, but it does not mean we cannot describe the dynamics in models such as Kirby's using the tools from evolutionary biology. The effects of induction in language change are a form of "directed mutation", and can be included, for instance, in the Price Equation in the transmission term. More work is needed to work this out with concrete examples.

11 Conclusions

In this paper I have discussed a variety of models from population genetics, evolutionary game-theory and social evolution theory. I have used these models to make a list of requirements for evolutionary scenarios of the biological and cultural evolution of language. These requirements correspond to the following questions we should ask when confronted with a scenario for the biological or cultural evolution of language:

- What are the units of inheritance the scenario assumes? Genes? Memes?
- What is the scope of variation in these genes or memes? That is, what is the assumed set of possible traits/strategies available for evolution?
- What are the selection pressures? That is, what is the assumed payoff for each of these possible traits in each possible context?
- For every innovation in the scenario, will it indeed be favoured by selection when extremely rare? If not, is there a non-negligible chance it could get established by stochastic effects, or get frequent enough to be favoured by selection?
- Does the assumed series of changes in the scenario indeed constitute a path of ever-increasing fitness? That is, is there a path of fit intermediates from start to finish?
- How much time will each of the innovations take to get established?
- Is there for every transition sufficient variation, but not too much?
- How does the scenario explain that speakers maintain the willingness to speak honestly, and that hearers continue to listen and believe the information received? That is, how does it solve the problem of cooperation?
- How does the scenario explain that speakers and hearers, after every innovation, agree on which signals refer to which meanings? That is, how does it solve the problem of coordination?
- How does the scenario relate dynamics at different levels of description – genes, strategies, individuals, groups, languages?

A Wright's Adaptive Topography

Consider the single locus, two alleles model of figure 1. Recall the expression for average fitness of the 3 possible genotypes (equation 5):

$$\overline{w} = p^2 w_{AA} + 2pq w_{Aa} + q^2 w_{aa} \tag{22}$$

Because $p + q = 1$ this expression can be rewritten as:

$$\begin{aligned} \overline{w} &= p^2 w_{AA} + 2p(1-p) w_{Aa} + (1-p)^2 w_{aa} \\ &= p^2 w_{AA} + 2p w_{Aa} - 2p^2 w_{Aa} + w_{aa} \\ &\quad -2p w_{aa} + p^2 w_{aa}. \end{aligned} \tag{23}$$

The derivative of $\overline{w}$ with respect to p is now (provided the fitness coefficients are independent of p):

$$\begin{aligned} \frac{\mathrm{d}\overline{w}}{\mathrm{d}p} &= 2p w_{AA} + 2w_{Aa} - 4p w_{Aa} - 2w_{aa} + 2p w_{aa} \\ &= 2\left(p w_{AA} + w_{Aa} - 2p w_{Aa} - w_{aa} + p w_{aa}\right) \\ &= 2\left(p w_{AA} + q w_{Aa} - p w_{Aa} - q w_{aa}\right) \\ &= 2\left(p\left(w_{AA} - w_{Aa}\right) - q\left(w_{aa} - w_{Aa}\right)\right). \end{aligned} \tag{24}$$

Now, recall the expression for the change in p (equation (6)), which can in a few steps be rewritten as:

$$\begin{aligned} \Delta p &= p' - p \\ &= \frac{p(p w_{AA} + q w_{Aa})}{\overline{w}} - p \\ &= \frac{p(p w_{AA} + q w_{Aa})}{\overline{w}} - \frac{p\overline{w}}{\overline{w}} \\ &= \frac{p}{\overline{w}}(p w_{AA} + q w_{Aa} - \overline{w}). \end{aligned} \tag{25}$$

Inserting equation (22) into equation (25), and rearranging using the fact that $q = 1 - p$, gives:

$$\begin{aligned} \Delta p &= \frac{p}{\overline{w}}(p w_{AA} + q w_{Aa} - p^2 w_{AA} - 2pq w_{Aa} - q^2 w_{aa}) \\ &= \frac{pq}{\overline{w}}(p(w_{AA} - w_{Aa}) - q(w_{aa} - w_{Aa})). \end{aligned} \tag{26}$$

Equation (26) and (24) can be combined into equation (9), as is explored in the main text.

References

BARTON, G. E. & BERWICK, R. C. (1987). *Computational Complexity and Natural Language*. Cambridge, MA: MIT Press.

BARTON, N. & PARTRIDGE, L. (2000). Limits to natural selection. *BioEssays* **22**, 1075–1084.

BARTON, N. & TURELLI, M. (1991). Natural and sexual selection on many loci. *Genetics* **127**, 229–255.

BARTON, N. & ZUIDEMA, W. (2003). Evolution: the erratic path towards complexity. *Current Biology* **13**, 649–651.

BATALI, J. (2002). The negotiation and acquisition of recursive grammars as a result of competition among exemplars. In: Briscoe (2002).

BRISCOE, T. (2000). Grammatical acquisition: Inductive bias and coevolution of language and the language acquisition device. *Language* **76**.

BRISCOE, T., ed. (2002). *Linguistic evolution through language acquisition: formal and computational models*. Cambridge University Press.

BRISCOE, T. (2003). Grammatical assimilation. In: Christiansen & Kirby (2003), pp. 317–337.

CAVALLI-SFORZA, L. & FELDMAN, M. (1983). Paradox of the evolution of communication and of social interactivity. *Proc. Nat. Acad. Sci. USA* **80**, 2017–2021.

CAVALLI-SFORZA, L. L. & FELDMAN, M. W. (1981). *Cultural Transmission and Evolution: A Quantitative Approach*. Princeton, NJ: Princeton University Press.

CHOMSKY, N. (1975). *Reflections on Language*. New York: Pantheon.

CHRISTIANSEN, M. H. (1994). *Infi nite Languages, Finite Minds: Connectionism, Learning and Linguistic Structure*. Ph.D. thesis, University of Edinburgh, Scotland.

CHRISTIANSEN, M. H. & KIRBY, S., eds. (2003). *Language Evolution*. Oxford, UK: Oxford University Press.

COYNE, J., BARTON, N. & TURELLI, M. (2000). Is Wright's shifting balance process important in evolution? *Evolution* **54**, 306–317.

CROW, J. F. (1999). Hardy, Weinberg and language impediments. *Genetics* **152**, 821–825.

DARWIN, C. (1859). *The Origin of Species – by means of natural selection or the preservation of favoured races in the struggle for life*. London: Murray. (this edition, New York: The New American Library, 1958).

DARWIN, C. (1871). *The Descent of Man, and selection in relation to sex*. London: John Murray. Reprinted in 1981 by Princeton University Press.

DAWKINS, R. (1976). *The Selfi sh Gene*. Oxford University Press. This edition 1989.

DAWKINS, R. (1982). *The Extended Phenotype*. Oxford: Oxford University Press.

DAWKINS, R. & KREBS, J. R. (1978). Animal signals: information or manipulation? In: *Behavioural ecology: an evolutionary approach* (Krebs, J. R. & Davies, N. B., eds.). Oxford, UK: Blackwell Scientific Publications.

DEACON, T. (1997). *Symbolic species, the co-evolution of language and the human brain*. The Penguin Press.

DESSALLES, J.-L. (1998). Altruism, status, and the origin of relevance. In: Hurford *et al.* (1998).

DUNBAR, R. (1998). Theory of mind and the evolution of language. In: Hurford *et al.* (1998).

EIGEN, M. (1971). Self-organization of matter and the evolution of biological macro-molecules. *Naturwissenschaften* **58**, 465–523.

FISHER, R. A. (1922). On the dominance ratio. *Proc Roy Soc Edin* **42**, 321–431.

FISHER, R. A. (1930). *The Genetical Theory of Natural Selection.*. Oxford, UK: Clarendon Press.

FITCH, W. T. (2004). Kin selection and "mother tongues": A neglected component in language evolution. In: *Evolution of Communication Systems: A Comparative Approach* (Oller, K. & Griebel, U., eds.), pp. 275–296. Cambridge, MA: MIT Press.

FRANK, S. A. (1998). *Foundations of Social Evolution*. Princeton University Press.

GARDNER, A. & WEST, S. (2004). Spite and the scale of competition. *Journal of Evolutionary Biology* in press.

GARDNER, R. & GARDNER, B. (1969). Teaching sign language to a chimpanzee. *Science* **165**, 664–672.

GOLD, E. M. (1967). Language identification in the limit. *Information and Control (now Information and Computation)* **10**, 447–474.

GRAFEN, A. (1979). The hawk-dove game played between relatives. *Animal Behaviour* **27**, 905–907.

GRAFEN, A. (1990). Biological signals as handicaps. *Journal of Theoretical Biology* **144**, 517–546.

GRAFEN, A. (2003). Fisher the evolutionary biologist. *Journal of the Royal Statistical Society: Series D (The Statistican)* **52**, 319–329.

HALDANE, J. B. S. (1932). *The causes of evolution*. New York: Longmans.

HAMILTON, W. (1964a). The genetical evolution of social behaviour. i. *Journal of Theoretical Biology* **7**, 1–16.

HAMILTON, W. (1964b). The genetical evolution of social behaviour. ii. *Journal of Theoretical Biology* **7**, 17–52.

HAMILTON, W. (1970). Selfish and spiteful behaviour in an evolutionary model. *Nature* **228**, 1218–20.

HARDY, G. H. (1908). Mendelian proportions in a mixed population. *Science* **28**, 49–50.

HARLEY, C. (1981). Learning the evolutionarily stable strategy. *Journal of Theoretical Biology* **89**, 611–33.

HAUSER, M. D. & FITCH, W. T. (2003). What are the uniquely human components of the language faculty? In: Christiansen & Kirby (2003), pp. 317–337.

HILL, W. & ROBERTSON, A. (1966). The effect of linkage on limits to artificial selection. *Genetical Research* **8**.

HINTON, G. E. & NOWLAN, S. J. (1987). How learning can guide evolution. *Complex systems* **1**, 495–502.

HURFORD, J. R. (1989). Biological evolution of the Saussurean sign as a component of the language acquisition device. *Lingua* **77**, 187–222.

HURFORD, J. R., STUDDERT-KENNEDY, M. & KNIGHT, C., eds. (1998). *Approaches to the evolution of language: social and cognitive bases*. Cambridge, UK: Cambridge University Press.

JOSHI, A., VIJAY-SHANKER, K. & WEIR, D. (1991). The convergence of mildly context-sensitive grammar formalisms. In: *Foundational issues in natural language processing* (Sells, P., Shieber, S. & Wasow, T., eds.), pp. 21–82. Cambridge MA: MIT Press.

KIRBY, S. (1994). Adaptive explanations for language universals: A model of Hawkins' performance theory. *Sprachtypologie und Universalienforshung* **47**, 186–210.

KIRBY, S. (1998). Fitness and the selective adaptation of language. In: Hurford *et al.* (1998).

KIRBY, S. (2000). Syntax without natural selection: How compositionality emerges from vocabulary in a population of learners. In: *The Evolutionary Emergence of Language: Social function and the origins of linguistic form* (Knight, C., Hurford, J. & Studdert-Kennedy, M., eds.). Cambridge, UK: Cambridge University Press.

KIRBY, S. (2002). Natural language from artificial life. *Artifi cial Life* **8**, 185–215.

KIRKPATRICK, M., JOHNSON, T. & BARTON, N. (2002). General models of multilocus evolution. *Genetics* **161**, 1727–50.

KOMAROVA, N., NIYOGI, P. & NOWAK, M. (2001). The evolutionary dynamics of grammar acquisition. *J. Theor. Biology* **209**, 43–59.

LAI, C., FISHER, S., HURST, J., VARGHA-KHADEM, F. & MONACO, A. (2001). A forkhead-domain gene is mutated in a severe speech and language disorder. *Nature* **413**, 519–23.

LEWIS, D. K. (1969). *Convention: a Philosophical Study*. Cambridge, MA: Harvard University Press.

LIEBERMAN, P. (1984). *The biology and evolution of language*. Cambridge, MA: Harvard University Press.

LILJENCRANTS, J. & LINDBLOM, B. (1972). Numerical simulations of vowel quality systems: the role of perceptual contrast. *Lan-*

guage **48**, 839–862.

MAYNARD SMITH, J. (1964). Group selection and kin selection. *Nature* **201**, 1145–1147.

MAYNARD SMITH, J. (1965). The evolution of alarm calls. *The American Naturalist* **99**, 59–63.

MAYNARD SMITH, J. (1982). *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, England.

MAYNARD SMITH, J. & HAIGH, J. (1974). The hitch-hiking effect of a favourable gene. *Genet.Res.* **23**, 23–35.

MAYNARD SMITH, J. & PRICE, G. R. (1973). The logic of animal conflict. *Nature* **246**, 15–18.

MAYNARD SMITH, J. & SZATHMÁRY, E. (1995). *The major transitions in evolution*. Oxford: W.H. Freeman.

MESOUDI, A., WHITEN, A. & LALAND, K. (2004). Perspective: is human cultural evolution darwinian? evidence reviewed from the perspective of the origin of species. *Evolution* **58**, 1–11.

MITCHENER, G. & NOWAK, M. A. (2002). Competitive exclusion and coexistence of universal grammars. *Bull Math Biol* **65**, 67–93.

NASH, J. F. (1950). Equilibrium points in n-person games. *Proc. Natl. Acad. Sci. USA* **36**, 48–49.

NIYOGI, P. (2002). Theories of cultural evolution and their applications to language change. In: Briscoe (2002).

NOWAK, M. A., KOMAROVA, N. & NIYOGI, P. (2001). Evolution of universal grammar. *Science* **291**, 114–118.

NOWAK, M. A. & KRAKAUER, D. C. (1999). The evolution of language. *Proc. Nat. Acad. Sci. USA* **96**, 8028–8033.

OKASHA, S. (2003). Biological altruism. In: *The Stanford Encyclopedia of Philosophy* (Zalta, E. N., ed.).

OLIPHANT, M. (1999). The learning barrier: Moving from innate to learned systems of communication. *Adaptive Behavior* **7**.

OLIPHANT, M. & BATALI, J. (1996). Learning and the emergence of coordinated communication. *Center for research on language newsletter* **11**, 1–46.

PARKER, G. A. & MAYNARD SMITH, J. (1990). Optimality theory in evolutionary biology. *Nature* **348**, 27–33.

PINKER, S. & BLOOM, P. (1990). Natural language and natural selection. *Behavioral and brain sciences* **13**, 707–784.

PRICE, G. R. (1970). Selection and covariance. *Nature* **227**, 520–521.

PRINCE, A. & SMOLENSKY, P. (2004). *Optimality Theory: Constraint interaction in generative grammar*. Blackwell.

PROVINE, W. (1986). *Sewall Wright and evolutionary biology*. Chicago, IL: University of Chicago Press.

VAN ROOIJ, R. (2004). Evolution of conventional meaning and conversational principles. *Synthese (Knowledge, Rationality and Action)* **139**, 331–366.

ROUGHGARDEN, J. (1979). *Theory of Population Genetics and Evolutionary Ecology: An Introduction*. New York: Macmillan. Reprinted 1987.

SAVAGE-RUMBAUGH, S., MCDONALD, K., SEVCIK, R. A., HOPKINS, W. D. & RUBERT, E. (1986). Spontaneous symbol acquisition and communicative use by pygmy chimpanzees (*Pan paniscus*). *Journal of Experimental Psychology: General* **115**, 211–235.

SEYFARTH, R., CHENEY, D. & MARLER, P. (1980). Monkey responses to three different alarm calls: evidence of predator classification and semantic communication. *Science* **210**, 801–803.

SEYFARTH, R. M. & CHENEY, D. L. (1997). Some general features of vocal development in nonhuman primates. In: *Social influences on vocal development* (Snowdon, C. T. & Hausberger, M., eds.), pp. 249–273. Cambridge, U.K.: Cambridge University Press.

SHERMAN, P. W. (1977). Nepotism and the evolution of alarm calls. *Science* **197**, 1246–1253.

SIEGELMANN, H. & SONTAG, E. (1991). Neural networks are universal computing devices. Tech. Rep. SYCON–91–08, Rutgers Center for Systems and Control.

SIMON, H. (1955). A behavioral model of rational choice. *Quarterly Journal of Economics* **69**, 99–118.

SIMON, H. (1969). *The Sciences of the Artifi cial*. Cambridge, MA: MIT Press. The Karl Taylor Compton lectures.

SKYRMS, B. (1996). *Evolution of the Social Contract*. Cambridge, UK: Cambridge University Press.

SMITH, K. (2004). The evolution of vocabulary. *Journal of Theoretical Biology* **228**, 127–142.

SZATHMÁRY, E. (1999). The first replicators. In: *Levels of Selection in Evolution* (Keller, L., ed.), pp. 31–52. Princeton, NJ: Princeton University Press.

TERRACE, H. S. (1979). *Nim*. New York: Knopf.

TOMASELLO, M. & BATES, E., eds. (2001). *Language Development: The Essential Readings*. Malden, MA: Blackwell.

TRAPA, P. E. & NOWAK, M. A. (2000). Nash equilibria for an evolutionary language game. *Journal of Mathematical Biology* **41**, 172–188.

TRIVERS, R. (1971). The evolution of reciprocal altruism. *Quarterly Review of Biology* **46**, 35–57.

WEINBERG, W. (1908). Über den nachweis der Vererbung beim Menschen. *Jahresh. Wuertt. Ver. vaterl. Natkd.* **64**, 369–382.

WILLIAMS, G. C. (1966). *Adaptation and Natural Selection*. Princeton, NJ: Princeton University Press.

WOLFRAM, S. (2002). *A New Kind of Science*. Champaign, IL: Wolfram Media.

WRIGHT, S. (1932). The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In: *Proceedings of the Sixth International Congress on Genetics*, pp. 355–366.

YAMAUCHI, H. (2001). The difficulty of the Baldwinian account of linguistic innateness. In: *Advances in Artifi cial Life (Proceedings 6th European Conference on Artifi cial Life, Prague)* (Kelemen, J. & Sosík, P., eds.), vol. 2159 of *Lecture Notes in Computer Science*, pp. 391–400. Berlin: Springer.

YANG, C. D. (2000). Internal and external forces in language change. *Language Variation and Change* **12**, 231–250.

ZAHAVI, A. (1975). Mate selection - a selection for a handicap. *Journal of Theoretical Biology* **53**, 205–214.

ZAHAVI, A. (1977). The cost of honesty (further remarks on the handicap principle). *Journal of Theoretical Biology* **67**, 603–605.

ZUIDEMA, W. (2003). How the poverty of the stimulus solves the poverty of the stimulus. In: *Advances in Neural Information Processing Systems 15 (Proceedings of NIPS'02)* (Becker, S., Thrun, S. & Obermayer, K., eds.), pp. 51–58. Cambridge, MA: MIT Press.

ZUIDEMA, W. & DE BOER, B. (2003). How did we get from there to here in the evolution of language? *Behavioral and Brain Sciences* **26**, 694–695.

ZUIDEMA, W. & HOGEWEG, P. (2000). Selective advantages of syntactic language: a model study. In: *Proceedings of the 22nd Annual Meeting of the Cognitive Science Society* (Gleitman & Joshi, eds.), pp. 577–582. Mahwah, NJ: Lawrence Erlbaum Associates.

An annotated bibliography of grammar induction models for natural language learning

Willem Zuidema
Institute for Logic, Language and Computation
University of Amsterdam
Plantage Muidergracht 24, 1018 TV, Amsterdam, the Netherlands.
jzuidema@science.uva.nl

May 23, 2008

1 Review Papers

Steven Pinker, 1979 *Pinker gives a good overview of the heuristic approaches for inducing context-free gramar existing at the time. He ridicules "automatated field linguists", discusses the Gold paradigm and argues it shows the impossibility of learning from plain text. He then discsses Andersons work, and argues for semantically informed bootstrapping.*

Dana Angluin and Carl H. Smith, 1983 , Inductive Inference: Theory and Methods, Computing Surveys 15,3:237-269. *Excellent review, surveying both abstract learnability results and concrete heuristic methods for learning grammars.*

Robin Collier, 1994 An Historical Overview of Natural Language Processing Systems that Learn. Artif. Intell. Rev. 8(1): 17-54 (1994) *Contains only a small section on syntax; most attention to algorithms for learning semantics and pragmatics.*

Yasubumi Sakakibara, 1997 Recent advances of grammatical inference, Theoretical Computer Science 185:15-45 *Good paper, but very much focused on Sakakibara's own work.*

R Parekh and V Honavar, 1998 Grammar Inference, Automata Induction, and Language Acquisition. In: Handbook of Natural Language Processing (Dale, Moisl & Somers). New York: Marcel Dekker. *Nice survey, starting with inference of finite-state automata and discussing stochastic and context-free extensions. Little new information on (P)CFGs.*

Colin de la Higuera, 2002 A Bibliographical Study of Grammatical Inference, unpublished. Extended version of: Colin de la Higuera (2000), Current trends in grammatical inference, In: Advances In Pattern Recognition, Lecture Notes In Computer Science 1876: 28-31.

Pieter Adriaans and Menno van Zaanen, 2004 Computational Grammar Induction for Linguists, Grammars, volume 7:57-68

2 Constituency-based Models

Ray Solomonoff, 1964: A Formal Theory of Inductive Inference, Part II, Information and Control, Part II: Vol. 7, No. 2, pp. 224-254. http://world.std.com/ rjs/1964pt2.pdf *This paper already presents a*

Bayesian approach to the identification of context-free grammars, such as later rediscovered by Cook et al and Stolcke. Solomonoff spells out how to define the prior probability distribution over grammars (a description length prior) and the likelihood of the data given the grammar.

Horning, 1969: Unpublished PhD-thesis; Horning defined a Bayesian objective function similar to Stolcke's (viewing grammars as being generated by a meta-grammar). The search procedure, however, is simply enumeration (like Gold (1967)). Horning proves that, if the input is generated stochastically, this algorithm identifies in the limit a correct grammar with probability 1 (i.e. stochastic CFGs are learnable).

Cook et al, 1976: Grammatical inference by hill climbing CM Cook, A Rosenfeld, AR Aronson - Informational Sciences (now: Information Sciences), 1976, 10:59-80. *presents a model very similar to that of Stolcke, 1994. It includes search operations equivalent to chunk and merge. The main difference is the objective function, where for both the grammar encoding (prior) and the data encoding (likelihood) terms, an additional costs is calculated for specifying the form of the rule/word. I.e. a grammar with rules* $\{X \rightarrow abbab, X \rightarrow babba\}$ *counts as less probable/more costly to encode than a grammar* $\{X \rightarrow aaaaa, X \rightarrow bbbbb\}$. *It is evaluated on a number of toy problems; Stolcke used these same problems and finds comparable performance.*

Wolff 1982, 1987: Wolff present a theory of child language development based on induction. He develops a number of computer programs to illustrate his point. They contain a lot of heuristics, but the essential operations in the model are incorporate, chunk and merge. The first, incorporates each sentence w1w2..wn into the grammar, by creating n unique nonterminal N1, N2, ..., Nn and adding rules S→N1 N2 .. Nn and N1→w1, N2→w2, ..., Nn→wn. Chunk then makes two consecutive nonterminals into a constituent. Merge equates two nonterminals (i.e. makes two words or constituents of the same category). The criterion for applying chunk and merge is based on the length of the grammar, measured in number of symbols at the right-hand side of rules. The algorithm iteratively searches for the pair of nonterminals that when chunked or merged leads to the greatest reduction in grammar length, and then applies it. Some of the additional heuristics have to do with fixing the greediness of the learning.

Langley, 1992; Langley & Stromsten, 2000: Reimplements a clean version of Wolff, now called "Grids", and uses it to argue for the usefulness of a "simplicity bias" in learning.

Adriaans 1992: an algorithm for inducing categorial grammars from plain text, called "Emile", later adapted to yield context free grammars.

Stolcke 1994: Stolcke develops a Bayesian approach to learning HMMs and PCFGs from plain text. The operations are the same as in Wolff and Langley, but the criterion to apply chunk and merge is different. It is based on an estimate of the posterior probability, which is itself the product of the likelihood of the data and the prior probability of the grammar. The prior is in turn the product of a structure prior (in the basic version exponentially decreasing with description length) and a parameter prior (peaking at uniform weights for all production rules). The likelihood is exponentially decreasing with the description length of the derivations of each of the observed sentences. Hence, in its basic version, the model corresponds to MDL learning, but Stolcke at some point changes the prior to bias towards slightly larger grammars. Stolcke makes a number of (rather crude) approximations to make the calculation of likelihood efficient: he assumes that most of the probability mass of a sentence is concentrated in its Viterbi parse (hence, for ambiguous sentences, only the probability of the most

probably parse is counted) and that the Viterbi property of a parse is preserved under the merge operation. There is no evaluation of how good these assumptions are. Finally, Stolcke uses a number of search strategies to find the grammars that maximize the posterior. The one that works best is a beam search for the best merges with a look-ahead to further improvements through merge, alternated with a beam search for the best chunks also with a look-ahead to further improvements through merge (chunks don't improve the posterior on their own). Stolcke only evaluates his algorithm on toy grammars, and shows he can easily find all of the examle grammars proposed by Langley and others.

Nevill-Manning and Witten, 1997: Nice paper, describing a compression algorithm called "Sequitur" for sequences (streams) of strings. It differs from the other algorithms discussed here, in that it doesnot use sentence boundaries but assumes a continuous stream of input. It finds common patterns in the input, replacing recursively frequent and long sequences with nonterminals, in the process creating hierarchical structure. By some clever encodings the algorithm operates in approximately linear time. It has been applied to large amounts of data, but cannot be evaluated in the same way as other algrithms.

Van Zaanen 2002: Van Zaanen develops a related but different model called Alignment-Based Learning. The algorithm has two phases: (1) alignment, (2) selection. In phase 1, it identifies all pairs of sentences that can be partially aligned (i.e. share subtrings). All of the *dissimilar* parts are then viewed as hypothesised constituents of the same category. If the same substring receives multiple labels by this process, these labels made equal. Thus, hypotheses on constituents are generated with a procedure that combines chunks and merges. The selection of a consistent set of constituent assignments is postponed until the phase 2. Here, the most probable set is selected, defined as the product of the relative frequencies (?) which, I'm guessing, comes down to selecting an implicit PCFG that maximizes the likelihood of the data (but of course constrained by the possible rules that phase 1 is generating). Van Zaanen evaluates his algorithm on ATIS and OVIS and, although performance is poor, it outperforms Adriaans' EMILE. The joint effect of the two phases makes it difficult to evaluate what the algorithm is optimizing.

Klein & Manning, 2002 EM based induction of binary branching constituent structure. This is the first paper to score better than the right-branching heuristic (unlabeled recall and precision on sentences from the WSJ corpus no longer than 10 words). The algorithm uses a chart representation, with just two possible values in each cell: constituent or distituent (i.e. a bracketing B). The likelihood of a sentence given such a chart, $P(S|B)$, is defined as the product of probabilities of each cell generating its yield and the corresponding context, conditioned on the constituency-value of the cell. (It seems to me that each word is generated multiple times in this model, by all cells that include it in their span, or consider it context to their span. The authors acknowledge the probability model is deficient in some way, but claim a fix is simple but cumbersome and irrelevant). The likelihood of a sentence is simply $P(S) = \sum_B P(B)P(S|B)$, with the prior $P(B)$ uniform over all binary branching trees. Using EM to maximize the likelihood of the whole corpus, the model find quite accurate bracketings of the sentences. On POS-tag sequences from WSJ10, the model scores $F_1 = 71.1\%$, substantially higher than the 60% of the right-branching heuristic.

Petasis et al 2003: Petasis presents a reimplementation of Langley's algorithm, but now formulated throughout as a MDL algorithm – removing, as he argues, some of Langley's inconsistencies on the way. The earlier version is called e-grids, a more recent one eg-grids. The most important innovation, as I view it, is not in the optimization criterion (MDL = Stolcke's posterior), but in the search strategy. In addition to a beam search for the right sequence for merges and chunks, Petasis proposes a number of

specific heuristics that perform a sequence of merges and chunks, and generate a range of candidate changes to the grammar to be selected in a second step. This allow him to find relevant linguistic patterns much more efficiently, thus hopefully avoiding local maxima that Stolcke's algorithm might get stuck in. Unfortunately, the algorithm has not been evaluated on any large corpora. In Borensztajn & Zuidema (2007), we evaluate a reimplementation of e-grids on the WSJ10 corpus.

Solan et al., 2005: In a series of papers, Solan et al study another related induction algorithm called ADIOS. The presentation is somewhat confusing, with non-standard terminology, alternative usage of established technical terms (such as "context-sensitive") and technical details spread over many different papers. The model also uses cooccurrence statistics to decide on likely constituents (here called "significant pattern") and syntactic category membership of those constituents (here called "equivalence classes"). The model starts with creating a huge graph with nodes for every word (type, not token), and sentences as labeled paths through that graph. The algorithm then finds those sequences of words that have a high "fan-in, fan-out" score. This is a local measure, measuring how often sentences follow the whole sequence relative to how often they branch off before, within or after the sequence. The algorithm continues iteratively, treating the found sequences as single words (constituents). This operation is, I think, identical to the chunking operation in Wolff and Stolcke (but with a different criterion for which words/nonterminals to apply it to). A second operation creates equivalence classes, essentially making sets of words or constituents substitutable (and hence of the same category). The application of this operation is also guided by a local measure.The model is said to outperform Adriaans and Van Zaanen on a small target grammar, and is further evaluated in various non-standard ways. An implementation with a strict limitation on the number of words can be downloaded from the website.

Bod, 2006ab: Radical approach, called U-DOP (for "Unsupervised Data-Oriented Parsing"), where all possible binary branching trees are assigned to the input sentences. All subtrees of all these trees than form a stochastic tree substitution grammar (STSG), with a weight proportional to the frequency in the subtree-multiset (in Bod 2006b subsequently reestimated using EM). The corpus is reparsed with (an approximation of) this STSG, and (an approximation of) the most probable parse for each sentence is determined. Excellent empirical results are reported, with the currently best bracketing score on POS-sequences from WSJ10: unlabeled $F_1 = 82.9$.

Peter Grünwald, 1996 A minimum description length approach to grammar inference, In: Symbolic, Connectionist and Statistical, Approaches to Learning for Natural Language Processing (eds. S. Wermter, E. Riloff, G. Scheler), Lecture Notes in Artificial Intelligence (Lecture Notes In Computer Science), 1040:203-216 http://www.cwi.nl/ pdg/ftp/mdlagi.ps. *Similar to Petasis et al, but somewhat preliminary.*

Alexander Clark, 2004 Grammatical Inference and the Argument from the Poverty of the Stimulus, AAAI Spring Symposium on Interdisciplinary Approaches to Language Learning, Stanford CA http://www.issco.unige.ch/staff/clark/SS704ClarkA.pdf

Jonas Kuhn, 2004 Experiments in Parallel-Text Based Grammar Induction, ACL'04. http://uts.cc.utexas.edu/ jonask/kuhn-acl04-final.pdf

Simon Dennis, 2005 An exemplar-based approach to unsupervised parsing. In: Bruno G. Bara, Lawrence Barsalou, and Monica Bucciarelli, editors, *Proceedings of the 27th Conference of the Cognitive Science Society*. Lawrence Erlbaum.

3 Dependency-based

Buszkowski & Penn, 1990 Algorithm for finding a classical categorical grammar from input data annotated with functor-argument structure. Lots of complicated math, but the algorithm doesn't actually seem to be able to learn interesting grammars.

Klein & Manning, 2004 Dan Klein and Christopher D. Manning, 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42th Annual Meeting of the ACL*.

Virginia Savova and Leon Peshkin, 2005: Bayesian Networks for the Recursive recovery of Syntactic Dependencies. In Proceedings of CogSci, 2005

Yoav Seginer, 2007: Fast Unsupervised Incremental Parsing. *Algorithm based on enriched dependency structures. Links between words are created based on co-occurrence statistics of words. These statistics are gathered on-line, which involves keeping track of the frequency of co-occurring words to the left and right, but also of the information that frequently co-occuring words carry about words they frequently co-occur with. This way, a word "knows" which other words have similar co-occurrence statistics. Currently best bracketing scores on WSJ10 (word sequences): unlabeled* $F_1 = 75.1$.

References

ADRIAANS, P. (1992). *Learning Language from a Categorical Perspective*. Ph.D. thesis, University of Amsterdam.

ANGLUIN, D. & SMITH, C. H. (1983). Inductive inference: Theory and methods. *Computing Surveys* **15**.

BOD, R. (2006a). An all-subtrees approach to unsupervised parsing. *Proceedings ACL-COLING'06* .

BOD, R. (2006b). Unsupervised parsing with U-DOP. In: *Proceedings of the 10th International Conference on Computational Natural Language Learning (CONLL-X)*.

BUSZKOWSKI, W. & PENN, G. (1990). Categorial grammars determined from linguistic data by unification. *Studia Logica* **49**, 431–454.

GOLD, E. M. (1967). Language identification in the limit. *Information and Control (now Information and Computation)* **10**, 447–474.

HORNING, J. (1969). *A study of grammatical inference*. Ph.D. thesis, Computer Science Dep., Stanford University.

KLEIN, D. & MANNING, C. D. (2002). A generative constituent-context model for improved grammar induction. In: *Proceedings of the 40th Annual Meeting of the ACL*.

LANGLEY, P. & STROMSTEN, S. (2000). Learning context-free grammars with a simplicity bias. In: *Proceedings of the Eleventh European Conference on Machine Learning*, pp. 220–228. Barcelona: Springer-Verlag.

NEVILL-MANNING, C. G. & WITTEN, I. H. (1997). Identifying hierarchical structure in sequences: A linear-time algorithm. *Journal of Artificial Intelligence Research* **7**, 67–82.

PETASIS, G., PALIOURAS, G., KARKALETSIS, V., HALATSIS, C. & SPYROPOULOS, C. (2004). E-grids: Computationally efficient grammatical inference from positive examples. *Grammars* **7**, 69–110.

PINKER, S. (1979). Formal models of language learning. *Cognition* **7**, 217–283.

SAKAKIBARA, Y. (1997). Recent advances of grammatical inference. *Theoretical Computer Science* **185**, 15–45.

SEGINER, Y. (2007). Fast unsupervised incremental parsing. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pp. 384–391. Prague, Czech Republic: Association for Computational Linguistics.

STOLCKE, A. (1994). *Bayesian Learning of Probabilistic Language Models*. Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, University of California at Berkeley.

STOLCKE, A. & OMOHUNDRO, S. M. (1994). Inducing probabilistic grammars by Bayesian model merging. In: *Proc. Second International Colloquium on Grammatical Inference and Applications (ICGI'94)*, vol. 862 of *Lecture Notes in Computer Science*, pp. 106–118. Berlin: Springer-Verlag.

WOLFF, J. G. (1982). Language acquisition, data compression and generalization. *Language & Communication* **2**, 57–89.