Article

Language *and* Speech

Three Design Principles of Language: The Search for Parsimony in Redundancy

Language *and* Speech
56(3) 265–290

DOI: 10.1177/0023830913484897
las.sagepub.com
SAGE

Barend Beekhuizen
Leiden University, the Netherlands

Rens Bod
University of Amsterdam, the Netherlands

Willem Zuidema
University of Amsterdam, the Netherlands

Abstract
In this paper we present three design principles of language – experience, heterogeneity and redundancy – and present recent developments in a family of models incorporating them, namely Data-Oriented Parsing/Unsupervised Data-Oriented Parsing. Although the idea of some form of redundant storage has become part and parcel of parsing technologies and usage-based linguistic approaches alike, the question *how much* of it is cognitively realistic and/or computationally optimally efficient is an open one. We argue that a segmentation-based approach (Bayesian Model Merging) combined with an all-subtrees approach reduces the number of rules needed to achieve an optimal performance, thus making the parser more efficient. At the same time, starting from *unsegmented wholes* comes closer to the acquisitional situation of a language learner, and thus adds to the cognitive plausibility of the model.

Keywords
Language acquisition, Data-Oriented Parsing, Bayesian Model Merging, learnability

Introduction: Three design principles of language

Modern linguistic theory has evolved along the lines of the principle of categoricity: knowledge of language is characterized by a categorical system of grammar. Numbers play no role (Bod, Hay, & Jannedy, 2003, p. 1). This idealization has been fruitful for some time, but it underestimates human

Corresponding author:
Barend Beekhuizen, Leiden University Centre for Linguistics, Leiden University, P.O. Box 9500, 2300 RA Leiden, the Netherlands.
Email: barendbeekhuizen@gmail.com

language capacities. There is a growing realization that linguistic phenomena at all levels of representation display properties of continua and show markedly gradient, probabilistic behaviour (see, e.g. Arnon & Snider, 2010; Bybee, 2006; Gahl & Garnsey, 2004; Tremblay & Baayen, 2010 – for discussion see also Gahl & Garnsey, 2006; Newmeyer, 2006).

An assumption often going hand in hand with the categoricity assumption is that linguistic knowledge is organized and represented mentally in an optimally parsimonious manner, namely by minimizing the amount of information encoded redundantly. Although this 'parsimony of representation' principle can yield important insights in organizational processes of the mind, it cannot be applied in an unconstrained manner, because language users do seem to produce and interpret language with what seem to be redundant structures with respect to their contents.

Finally, many current linguistic theories adhere to some form of a 'principle of homogeneity', assuming repertoires of building blocks (phones, words, rules), of roughly similar shape and size.

In this paper, we argue in favour of a family of models that violates all three of these principles: Data-Oriented Parsing (DOP; Bod, 1998; Scha, 1990). DOP includes probabilistic processing and learning models that allow fragments of many different shapes and sizes, partially overlapping with each other, to become the building blocks of language; DOP relaxes the constraints on representations dictated by the principles of categoricity, parsimony and homogeneity, while greatly reducing the complexity of the learning mechanism.

Our aim is not just to account for gradient linguistic phenomena at the 'computational' level (in Marr's (1982) hierarchy of modelling approaches); rather, in this paper, we investigate the applicability of DOP as a cognitive model at Marr's 'processing' level. In the following we will first briefly discuss three alternative design principles: experience, heterogeneity and redundancy. We then briefly review how DOP implements those principles, discuss recent extensions of DOP to unsupervised learning and end with some new directions in DOP-style models of language acquisition.

1.1 Messy, spurious and redundant

Following work by Spivey and Tanenhaus (1998), Gahl and Garnsey (2004), Bresnan (2007) and many earlier arguments (see Bod et al., 2003, for an overview and historical references), we view probabilities as an inherent part of the human language system. The language processing system is set up in such a way that, whenever an instance of a linguistic structure is processed, it is seen as a piece of evidence that affects the structure's probability distribution.

Probabilistic models of language, however, are not just about 'performance' either, as a topic that could be studied separately from linguistic 'competence'. The inseparability of probabilities from competence has become most obvious through the study of wide-coverage parsing automatic grammatical analysis that should be able to handle any sentence in a given language (Manning & Schütze, 1999). Research in this field has demonstrated that grammars that are general enough to cover the vast majority of sentences become so permissive that they license a prohibitive number of analyses for every sentence and deem almost any word sequence grammatical. Hence, even to account for the data dearest to the theoretical linguists in the categorical tradition, weights or probabilities are indispensable (Abney, 1996). Our first principle, therefore, is the principle of experience:

> **Principle 1 (Experience):** Knowledge of language is sensitive to distributions of previous language experiences. An adequate model of language processing should, whenever an expression is processed, treat it as a piece of evidence that affects the probability distributions over

linguistic structures. New expressions are constructed by probabilistically generalizing over previous expressions.

The assumption of the redundancy of representation[1] also goes beyond developing a modelling technique. We claim that language users keep track of complex structures that might be built up from smaller known structures, because some part of their application (e.g. their frequency or their specific meaning) cannot be reduced to the composition of the smaller known structures. While many linguists agree that there is a need to integrate probabilities into linguistics, the question where probabilities are represented is often still answered with reference to the principle of parsimony of representation. In this paper, the answer to the question where probabilities are represented is: Everywhere. Probabilities are relevant at all levels of representation, from phonetics and syntax to semantics and discourse and at all levels of abstraction: from complete, holistic Gestalts to the minimal building blocks.

In syntactic structure, for example, there is evidence that the units that carry probabilities are of many different sizes and shapes. For example, suppose that we want to produce a sentence corresponding to a meaning of asking someone's age. There may be several sentences with such a meaning, such as *How old are you?, What age do you have?* or even *How many years do you have?* Yet the first sentence is more acceptable than the other ones in that it corresponds to the conventional way of asking someone's age in English. Note that the difference is not due to the probabilities of the individual words, context-free productions or even the construction *How* ADJ *are you?*, where ADJ could be filled with any adjective. This is clear from the fact that the conventional way to ask about another personal attribute, weight, is not *How heavy are you?* It really seems that *How old are you?* is stored as a unit, and carries a weight as a unit. Evidence for an analysis is easily found in a representative corpus of English.

There are many more examples in the Construction Grammar and related literature of multi-word units that must be assumed to be stored as units in the language user's brain (see, e.g. Culicover & Jackendoff, 2005). Importantly, they range from complete sentences to two-word combinations, from semantically and syntactically completely regular (such as above), to completely idiosyncratic (such as *by and large*), and from completely lexically specific, to quite abstract (such as V POSS *way* PP, as in *He cheated his way out of jail* – see Verhagen, 2005). Our second design principle is therefore:

Principle 2 (Heterogeneity): An adequate model of human language processing must allow for a heterogeneous store of elementary units, ranging from single words, and basic combinatory rules, to multi-word constructions with various open slots and complete sentences.

The principles of experience and heterogeneity lead to our third design principle, which can be seen as a logical consequence of the first two, but is nevertheless surprising to many theorists and important enough to mention separately: redundancy. We argued that *How old are you?* is likely to be stored as a unit, but it is also perfectly regular and there is no denying that all of its component parts can perfectly function as independent units and combine with other elements. This implies that the exact same sentence, with the same grammatical structure, can in fact be derived in multiple ways (this is different from structural ambiguity, where different derivations assign different grammatical structure to a sentence).

Sometimes, such alternative derivations will correspond to differences in meaning, such as in *she kicked the bucket*, where a derivation that takes *kick the bucket* as a unit also carries its idiomatic meaning ('to die'), while a derivation that starts from the individual words will arrive at the

literal meaning. All syntactic theories allow for multiple items for the same words to be stored in the lexicon to account for this.

However, we will go a step further and also allow for multiple (in fact, many) alternative derivations of the same sentence, with the same structure and the same semantics. In computational linguistics, such inconsequential ambiguity is sometimes termed spurious ambiguity. Note that, by its very nature, it is logically impossible to provide semantic or syntactic evidence for or against spurious ambiguity. In theoretical linguistics it is typically ruled out with an appeal to parsimony of representation. However, representational parsimony comes at a cost: an increase in the complexity of the mechanisms needed to process, acquire, recognize and produce linguistic structures. We argue instead for a parsimony of mechanism. By allowing for a massively redundant storage of linguistic structures, we can account for language use and acquisition using surprisingly simple mechanisms, as we will show in the remainder of the paper.

Outside of the usual sources of evidence for theoretical linguistics, there is in fact plenty of evidence for the redundancy that we assume. A first source of evidence is historical linguistics. A well-attested example of syntactic change is the case of *kind of* constructions in English (Hopper & Traugott, 2003). *Kind* obviously started off as a noun, but gradually got reanalysed, jointly with of as an adjective and sometimes even as an adverb, as in *I kind of wanna go home*. As *kind of* cannot be a noun in this sentence, two changes must have occurred simultaneously: a reanalysis of the constituent-structure (where *kind* and *of* are joined), and a recategorization (from noun to adverb). Such changes are much easier accounted for when we assume a model where the subtree $_{\text{NP}}$[$_{\text{N}}$[*kind*] $_{\text{PP}}$[$_{\text{PREP}}$[*of*] NP]] is already a permissible building block even before the reanalysis, so that reanalysis only has to apply to a single complex whole as opposed to a chain of derivative operations. Other evidence comes from psycholinguistics, where it is found that reaction times on a grammatical judgement tasks are influenced by the frequency of occurrence of any substring of the test sentence (up to length 4), strongly suggesting there is at least a memory trace of strings of any length (cf., Arnon & Cohen-Priva, this issue; Bannard & Matthews, 2008; Bod, 2001; Tremblay & Baayen, 2010). We include such redundancy as a design feature of our model:

> **Principle 3 (Redundancy):** An adequate model of human language processing must be massively redundant, allowing for many different derivations of the same sentence, with many derivations yielding the same grammatical and even semantic structure.

We thus arrive at three related propositions about models of human language processing that run counter to central (although often implicit) assumptions in theoretical linguistics: the assumptions of categoricity, homogeneity and parsimony. Our approach, in contrast, emphasizes that the language faculty is *probabilistic, heterogeneous* and *redundant*. This might upset theorists that have, with Chomsky (1957, p. 48), hoped for linguistic theory to eventually uncover an elegant underlying system of regularities like the Periodic Table in chemistry. We believe much of language has turned out to be fundamentally erratic, messy and spurious, but we will show that one can nevertheless build elegant models that take these facts of language into account.

2 Exploiting redundancy: Data-Oriented Parsing

A first approach that takes the direct consequence of the principles given above is DOP (Bod, 1992, 1998; Bod, Scha, & Sima'an, 2003; Goodman, 2003; Kaplan, 1996; Sangati & Zuidema, 2011; Scha, 1990; Sima'an, 1996; Zuidema, 2007; a.o.). This approach analyses and produces

new sentences by combining fragments from previously analysed sentences stored in a 'corpus'. Fragments can be of arbitrary size, ranging from simple context-free rules to entire trees, thereby allowing for both productivity and idiomaticity. The frequencies of occurrence of the fragments are used to compute the distribution of most probable analyses for a sentence (in perception), or the distribution of most probable sentences given a meaning to be conveyed (in production).

2.1 DOP's building blocks: Treelets of arbitrary size and shape

What does such a general model, which takes all fragments from previous data, look like? In this section, we will illustrate a DOP model for syntactic surface constituent trees, although we could just as well have illustrated it for phonological, morphological or other kinds of representations. Consider a corpus of only two sentences with their syntactic analyses given in Figure 1 (we leave out some categories to keep the example simple).

On the basis of this corpus, the (new) sentence *She saw the dress with the telescope* can, for example, be derived by combining two fragments from the corpus – which we shall call *treelets* or *subtrees* – as shown in Figure 2. Note that there is no explicit distinction between words and structure in the subtrees. The combination operation between subtrees will for our illustration be limited to *label substitution*. This operation, indicated as °, identifies the leftmost non-terminal leaf node of the first subtree with the root node of the second subtree. That is, the second subtree is substituted on the leftmost non-terminal leaf node of the first subtree provided that their categories match.

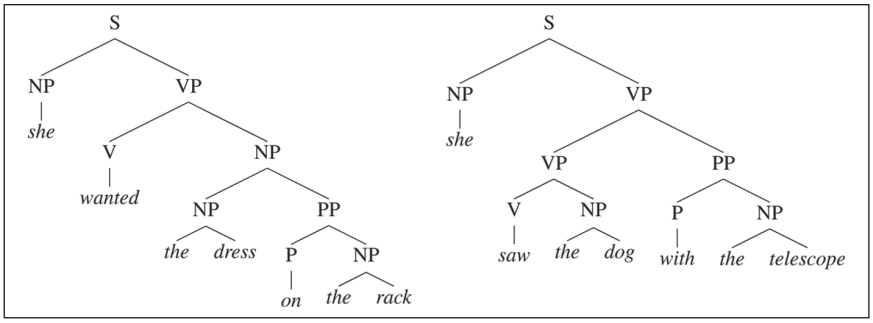

Figure 1. An extremely small corpus of two phrase-structure trees.

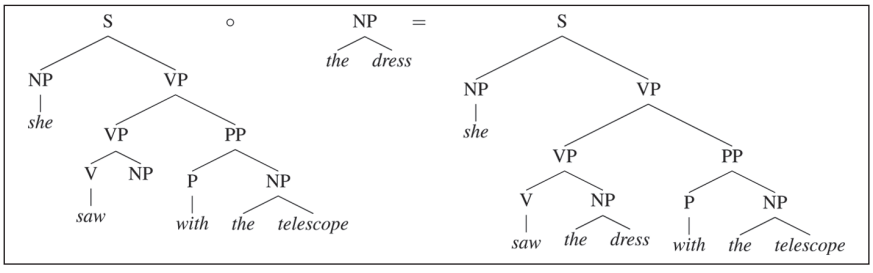

Figure 2. Analysing a new sentence by combining subtrees from Figure 1.

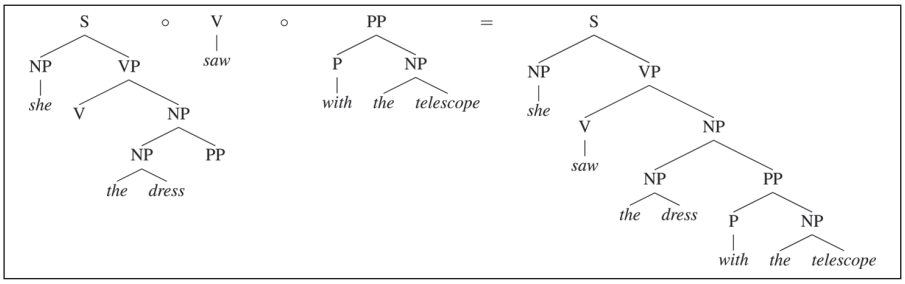

Figure 3. A different derivation for *She saw the dress with the telescope*.

Thus in Figure 2, 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 1, as shown in Figure 3. We will write $(t \circ u) \circ v$ as $t \circ u \circ v$ with the convention that $\circ$ is *left*-associative.

In Figure 3, the sentence *She saw the dress with the telescope* is analysed 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 2. Thus, the sentence is ambiguous in that it can be derived in (at least) two different ways, being either analogous to the first tree or to the second one in Figure 1.

Note that an unlimited number of sentences can be generated by combining subtrees from the corpus in Figure 1, such as *She saw the dress on the rack with the telescope, She saw the dress with the dog on the rack with the telescope*, etc. Thus we obtain unlimited productivity by finite means. Formally, this instance of DOP is equivalent to a Tree-Substitution Grammar (see Bod, 1998, and also Post & Gildea, this issue, and references therein).

2.2 How to enrich DOP with probabilities: The original approach

By having defined a method for combining subtrees from a corpus of previous trees into new trees, we effectively established a way to view a corpus as a tree generation process. This process becomes a stochastic process if we take the frequency distributions of the subtrees into account. For every tree and every sentence we can compute the probability that it is generated by this stochastic process. Let us illustrate the generation process by means of an even simpler corpus. Suppose that our example corpus consists of the two phrase-structure trees in Figure 4.

To compute the frequencies of the subtrees in this corpus, we need to define the (multi)set of subtrees that can be extracted from the corpus trees, which is given in Figure 5.

As explained above, by using the substitution operation, new sentence analyses can be constructed by means of this subtree collection. For instance, an analysis for the sentence *Mary likes Susan* can be generated by combining the three subtrees in Figure 6 from the set in Figure 5.

For the following it is important to distinguish between a *derivation* and an *analysis* (or *parse tree*) of a sentence. By a derivation of a sentence we mean a sequence of subtrees the first of which has its root labelled with S and for which the iterative application of the substitution operation produces the particular sentence. By an analysis of a sentence we mean the resulting parse tree of a derivation of the sentence. Then the probability of the derivation in Figure 6 is the *joint* probability of three stochastic events: (1) selecting the subtree $_{S}[\text{NP }_{VP}[\ _{V}[\textit{likes}]\ \text{NP}]]$ among the bag of subtrees with root

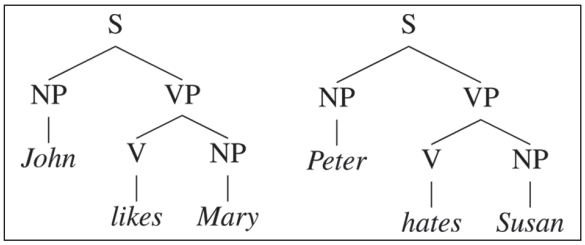

Figure 4. A corpus of two trees.

label S; (2) selecting the subtree $_{NP}$[*Mary*] among the subtrees with root label NP; and (3) selecting the subtree $_{NP}$[*Susan*] among the subtrees with root label NP. The probability of each event can be computed from the frequencies of the occurrences of the subtrees in the corpus. For instance, the probability of event (1) is computed by dividing the number of occurrences of the subtree $_{S}$[NP $_{VP}$[$_{V}$[*likes*] NP]] by the total number of occurrences of subtrees with root label S: 1/20.

In general, let | *t* | be the number of times subtree *t* occurs in the bag and *r*(*t*) be the root node category of *t*, then the probability assigned to *t* is

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

Since in our stochastic generation process each subtree selection is independent of the previous selections, the probability of a derivation is the product of the probabilities of the subtrees it involves. Thus, the probability of the derivation in Figure 6 is 1/20 × 1/4 × 1/4 = 1/320. In general, the probability of a derivation $t_1 \circ \ldots \circ tn$ is given by

$$P(t_1 \circ \ldots \circ t_n) = \Pi_i P(t_i)$$

The probability of an analysis or parse tree is *not* equal to the probability of a derivation producing it. There can be many different derivations resulting in the *same* parse tree. Although redundant from a linguistic point of view, from a statistical point of view these spurious derivations really matter: all derivations resulting in a certain parse tree contribute to the probability of that tree.

The probability of a parse tree is the probability that it is produced by any of its derivations. That is, the probability of a parse tree *T* is the sum of the probabilities of its distinct derivations *D*:

$$P(T) = \sum_{D \text{ derives } T} P(D)$$

Analogous to the probability of a parse tree, the probability of an utterance is the probability that it is yielded by any of its parse trees. For the task of language comprehension, we are often interested in finding the most probable parse tree given an utterance – or its most probable meaning if we use a corpus in which the trees are enriched with logical forms – and for the task of language production we are usually interested in the most probable utterance given a certain meaning or logical form.

There tends to be a preference in DOP *for the parse tree that can be constructed out of the largest possible corpus fragments* (Bod, 2006a), and thus for the parse tree that is most similar to previously seen utterance analyses. The same holds for the probability of an utterance: there is a preference for the utterance (given a certain meaning or intention) that can be constructed out of

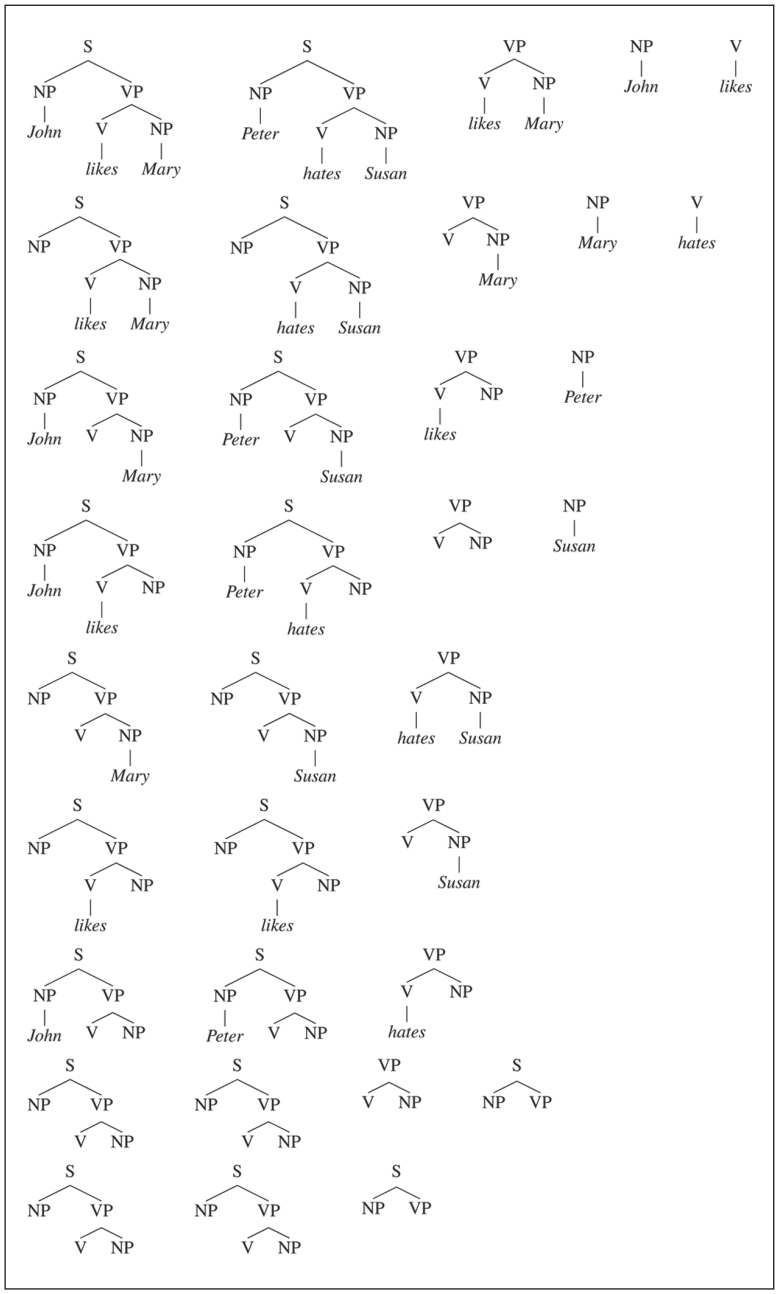

Figure 5. The bag of subtrees derived from the trees in Figure 4.

the largest possible corpus fragments, thus being most similar to previously seen utterances. This feature is important for explaining the use of constructions and prefabricated word combinations by natural language users.

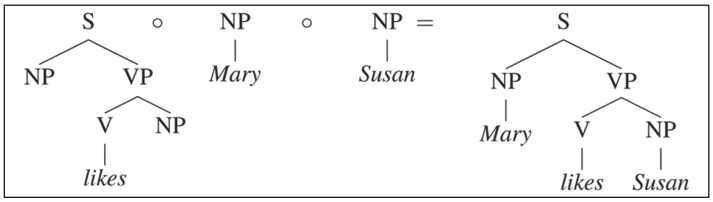

Figure 6. Analysing *Mary likes Susan* by combining subtrees.

The probability model explained here is one of the simplest probability estimators for DOP, better known as 'DOP1'. For an overview of different probability models for DOP, see for instance Zuidema (2006). During the last few years, a rich literature on DOP has emerged, yielding state-of-the-art results on the Penn treebank benchmark (Bansal & Klein, 2010; Bod, 2009; Sangati & Zuidema, 2011) and inspiring developments in related frameworks, including tree kernels (Collins & Duffy, 2001), reranking (Charniak & Johnson, 2005) and Bayesian adaptor and fragment grammars (e.g. Cohn, Goldwater, & Blunsom, 2009; Johnson, Griffiths, & Goldwater, 2007; Post & Gildea, 2009, this issue; O'Donnell, 2011).

2.3 Extending DOP to unsupervised learning

A limitation of the DOP approach is that it needs a corpus of already analysed sentences to start with. This means that DOP can at best model adult language processing. A first step to extend the approach towards language learning was proposed by Bod (2006b, 2007), where it is called '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 lets linguistic experience decide which is the most likely one. As a first approximation we will limit the set of all possible trees to unlabelled binary trees.[2] Conceptually, we can distinguish three learning phases under U-DOP:

(i) assign all possible (unlabelled binary) trees to a set of given sentences;
(ii) divide the binary trees into all subtrees;
(iii) compute the most probable tree for each sentence.

The only prior knowledge assumed by U-DOP is the notion of tree and the concept of most probable tree. U-DOP thus inherits the rather agnostic approach of DOP: We do not constrain the units of learning beforehand, but take all possible fragments.[3] We will discuss below how such an approach generalizes to other learning models, but we will first explain U-DOP in some detail.

2.3.1 Assign all unlabelled 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. The set of all unlabelled binary trees for the sentences *watch the dog* and *the dog barks* is given in Figure 7, 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 labels to the words, but this can be done as well.

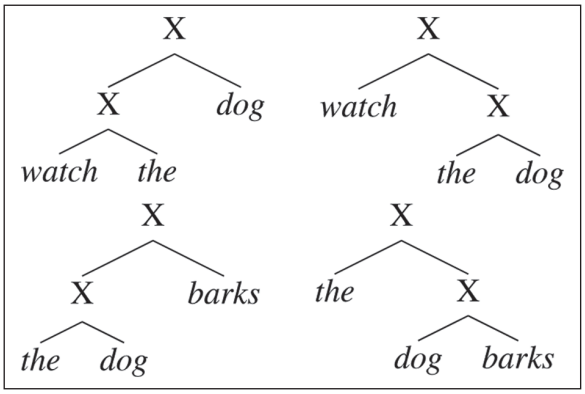

Figure 7. The unlabelled binary tree set for *watch the dog* and *the dog barks*.

2.3.2 Divide the binary trees into all subtrees. Figure 8 lists the subtrees that can be extracted from the trees in Figure 7. The first subtree in each row represents the whole sentence as a chunk, while the second and the third are 'proper' subtrees. Note that while most subtrees occur once, the subtree $_{X}$[*the dog*] occurs twice.

2.3.3 Compute the most probable tree for each sentence. From the subtrees in Figure 8, U-DOP can compute the most probable tree 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 8, two phrase-structure trees can be generated by U-DOP for this sentence, shown in Figure 9. Both tree structures can be produced by two different derivations, either by trivially selecting the largest possible subtrees from Figure 8 that span the whole sentence or by combining two smaller subtrees.

Thus 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. U-DOP computes the most probable tree in the same way as the supervised version of DOP explained above. Since the subtree $_{X}$[*the dog*] is the only subtree that occurs more than once, we can informally predict that the most probable tree corresponds to the structure $_{X}$[$_{X}$[*the dog*] *barks*], where *the dog* is a constituent. This can also be shown formally by applying the probability definitions given in Section 2:

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

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

Thus the tree $_{X}$[$_{X}$[*the dog*] *barks*] 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. 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).

For the sake of simplicity, we only used trees without lexical categories in our illustration of U-DOP. However, we can straightforwardly assign abstract labels *X* to the words as well. If we do

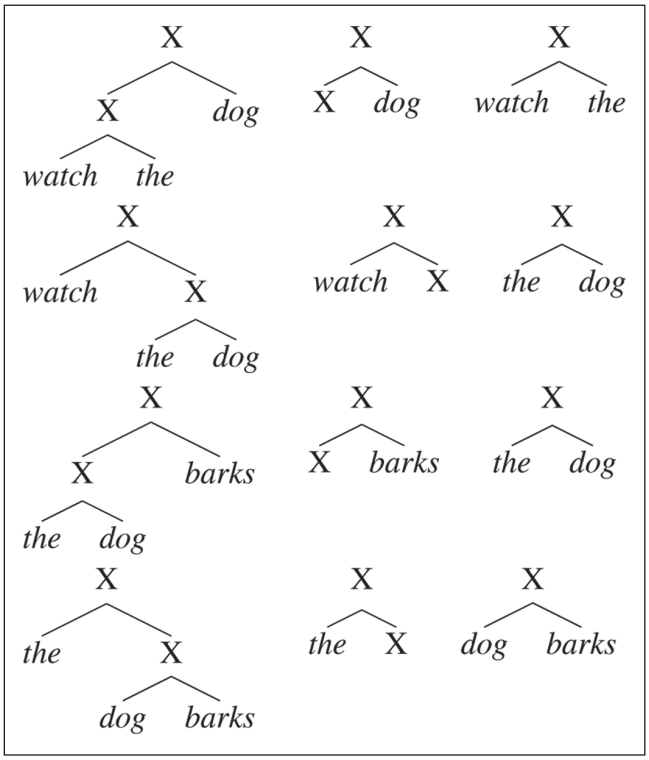

Figure 8. The subtree set for the binary trees in Figure 7.

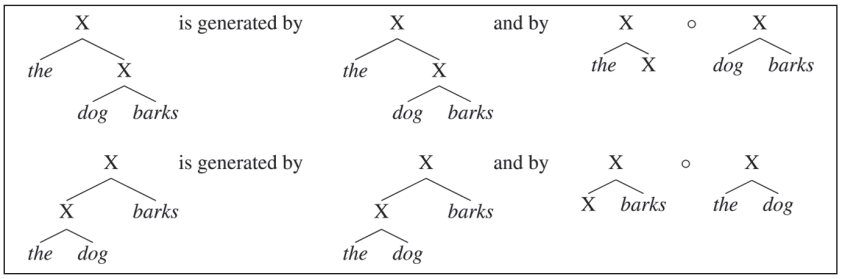

Figure 9. Parsing *the dog barks* from the subtrees in Figure 8.

so for the sentences in Figure 7, 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 refer to as a *discontiguous* subtree.

Discontiguous subtrees are important for covering a range of linguistic constructions, as those given in italics in sentences (1)–(5):

(1) BA carried *more* people *than* cargo in 2005.
(2) *What's* this scratch *doing* on the table?

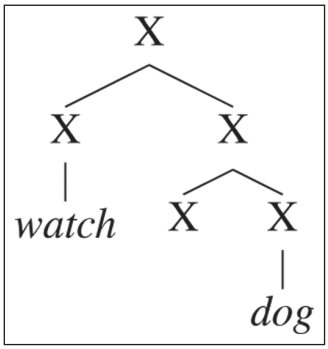

Figure 10. A discontiguous subtree.

(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, for example, (1)–(3), and particle verbs, for example (4), to regular syntactic agreement phenomena as in (5). The notion of subtree can easily capture the syntactic structure of these discontiguous constructions.

2.4 U-DOP's solutions to nativist puzzles

An important characteristic of U-DOP is that it can learn complex syntactic facets that are typically assumed to be governed by innate rules or constraints. Instead, in U-DOP/DOP rule-like behaviour can be a side-effect of computing the most probable analysis. To show this we will discuss with some detail the phenomenon of auxiliary fronting. This phenomenon, although originally introduced to argue for 'structure dependence' alone, 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 us start with the usual examples, which are the same as those used by 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 favour 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 & Nakayama, 1987).

(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 & Nakayama, 1987). On the other hand, it has been claimed that this type of sentence can be learned from experience alone (Lewis & Elman, 2001; Perfors, Tenenbaum, & Regier, 2011; Reali & Christiansen, 2005). We will not enter the controversy at this point (see Kam, Stoyneshka, Tornyova, Fodor, & Sakas, 2008), but believe that both viewpoints overlook an alternative possibility, namely that auxiliary fronting needs neither be innately specified nor in the input data in order to be learned. Instead, the phenomenon may be a side-effect of computing the most probable sentence structure without learning any explicit rule or constraint for this phenomenon.

The learning of auxiliary fronting can proceed when we have induced tree structures for the following two sentences:

(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

It can be shown that the most probable trees for (10) and (11) computed by U-DOP from sentences (10)–(15) are those in Figure 11 (see Bod, 2009, for details).

Given these trees, we can easily show that the most probable tree produces the correct auxiliary fronting. 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 12 from the acquired structures in Figure 11 (note that the first subtree is discontiguous).[4]

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 11, which are given in Figure 13.

The derivation in Figure 12 turns out to be the most likely one, thereby overruling the incorrect form produced Figure 13. This may be intuitively understood as follows: We have already explained in Section 2.3 that (U-)DOP's probability model has a very strong preference for sentences and structures that can be constructed out of the largest corpus fragments. This means that sentences generated by a shorter derivation tend to be preferred over sentences that can only be generated by longer derivations.

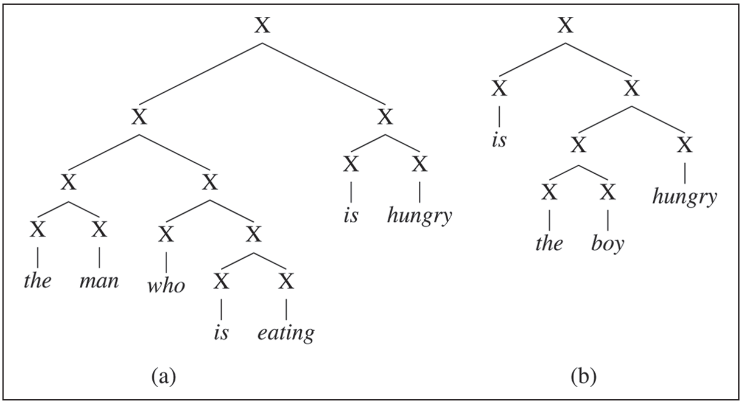

Figure 11. Tree structures for *the man who is eating is hungry* and *is the boy hungry?* learned by U-DOP from sentences (10)–(15).

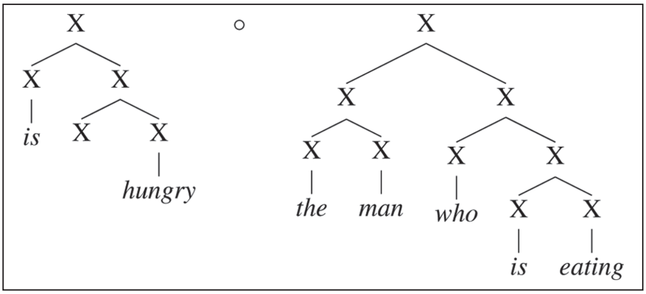

Figure 12. Producing the correct auxiliary fronting by combining two subtrees from Figure 11.

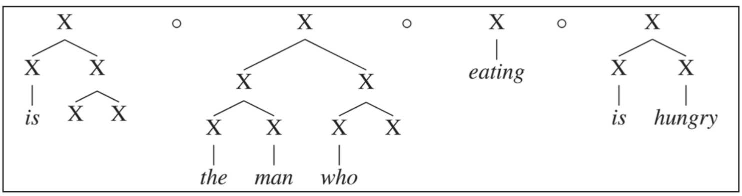

Figure 13. Producing the incorrect AUX-fronting by combining four subtrees from Figure 11.

We should keep in mind that the example above is limited to a couple of artificial sentences. Bod (2009) showed that U-DOP can also learn auxiliary fronting from the utterances in the Eve corpus (MacWhinney, 2000), even though complex auxiliary fronting does not occur in that corpus. In a similar vein, Bod and Smets (2012) applied U-DOP to other linguistic phenomena thought to be governed by constraints and rules. For, amongst other things, the Complex NP-constraint, Subject WH-questions, WH-questions in situ, superiority effects of question words, the lack of auxiliary fronting in embedded WH-questions and the blocking of movement from WH-islands, they showed that given a phrase-structure grammar, trained in an unsupervised fashion on child-directed speech (the Adam corpus in the CHILDES database), the model correctly assigned more

probability mass to grammatical variants of the constructions than the allegedly blocked ungrammatical ones. This suggests that these patterns can be predicted on the distributional information in the input data, although the patterns themselves need not be observed directly. Ungrammatical variants appear to be less probable than their grammatical counterparts. Bod and Smets argue that a simple unsupervised learning model such as U-DOP can solve virtually all nativist 'puzzles' pertaining to syntactic ordering phenomena reported in the literature without assuming separate constraints (see also Perfors et al., 2011, for similar arguments). Moreover, in several cases it turned out that only little syntactic structure was needed: the linguistic phenomena could often be learned by simply combining previously encountered sequential phrases (see Frank, Bod, & Christiansen, 2012, for further discussion).

3 Current developments: Principled constraints on redundancy

In U-DOP, the learner considers all binary subtrees, from minimal depth-one rules to full projections over an entire utterance. Taking our principles of experience and heterogeneity seriously, some aspects of U-DOP are problematic from the principles of experience and heterogeneity. In this section, we discuss these conceptual problems and take up the challenge of developing a developmentally plausible model.

In U-DOP, subtrees consist of binary branching trees (such as $_{X}$[X X]) and lexical productions (such as $_{X}$[*dog*]). These are given by U-DOP as the basic building blocks. This starting point seems to contradict two of the principles laid out in Section 1. Given the principle of *heterogeneity*, the constraint of binarity seems unwarranted. Perhaps language users produce and perceive wider fragments. A learner driven by *experience* would indeed have to figure out what kind of widths trees might have without having a hard-wired deterministic bias towards binary branchings. More pressingly, the starting point of binary branching trees forces the learner to conceptualize the initial analysis of every utterance as *hierarchical*, whereas it might be the case that learners do not come equipped with this preconception, but rather induce the fact that language structure is hierarchical from the nature of the data.

In this section, we explore a model that can be used to induce hierarchical structure, namely a variant of Bayesian Model Merging (BMM; Stolcke & Omohundro, 1994). BMM consists of two steps: firstly, the data (i.e. strings of words) are incorporated as completely flat rules in the grammar. Then the model tries to arrive at a grammar that optimally balances the compactness of its representation (a more abstract grammar can be encoded more compactly) and the fit with the data (a more abstract grammar fits the data less precisely). It does so by replacing substrings of the right-hand sides of the rules by novel symbols, effectively introducing more hierarchical analyses and novel constituent classes.

3.1 Segmentation of initial analyses with Bayesian Model Merging

The approach we take differs on several points from BMM as proposed by Stolcke and Omohundro (1994). We will encounter the differences as we go. Suppose we have the two sentences *John likes Mary* and *John likes Fred.* Using U-DOP, we can extract 20 subtrees from these examples. Several of those, however, are not observed by the learner to be used in any productive way, such as the ones in Figure 14. Therefore, the learner has little evidence to analyse these subtrees as independent units. Rather, it suffices to have a $_{X}$[*John likes* X] fragment and two fragments stating that *Mary* constitutes an X and so does *Bill*. Let us call these *minimal fragments*. A minimal fragment is a depth-one, *n*-ary (for $n > 0$) tree where the leaf nodes can be lexical items or slots (X-labels).

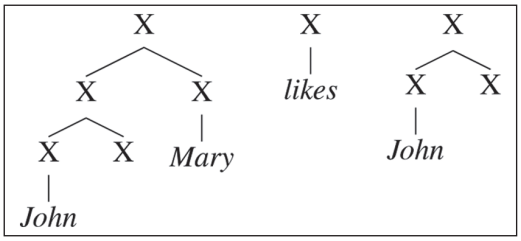

Figure 14. Three unproductive fragments on the basis of the sentences *John likes Mary* and *John likes Fred*.

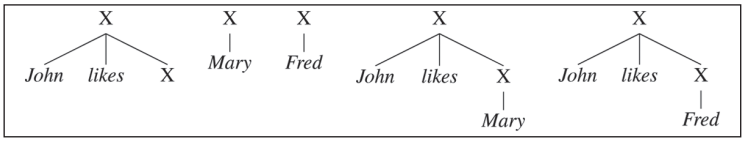

Figure 15. An all-subtrees grammar on the basis of the minimal fragments in the training items *John likes Mary* and *John likes Fred*.

For each training item, these minimal fragments constitute a parse tree of that training item. The minimal fragments $_{X}$[*John likes* X] and $_{X}$[*Mary*] can be composed into the parse tree $_{X}$[*John likes* $_{X}$[*Mary*]]. On the basis of these parse trees we can derive an all-subtree grammar such as the one in Figure 15. A *subtree* is then a subgraph of a parse tree, containing one or more minimal fragments.

Now, how does a learner arrive at the specific set of minimal fragments underlying the set of subtrees in Figure 15? Why would this segmentation be better than having no segmentation (i.e. each training item consisting of one big minimal fragment) or a situation in which the analysis consists of two binary minimal fragments and three word production fragments of the type $_{X}$[*likes*]? Intuitively, the former situation seems to lack a generalization, whereas the latter makes an unwarranted abstraction. Thus, we need a mechanism to determine what amount of abstraction and hierarchy over the training items is optimal.

We try to find this optimum, like BMM, by rewriting the elements of the grammar, in our case the minimal fragments. This search has several steps. By comparing two minimal fragments and rewriting them in a more compact format, the model tries to arrive at an optimal trade-off between the compactness of the grammar and the degree to which this analysis fits the data. The model starts with a fully unsegmented grammar (i.e. all minimal fragments are of the form $_{X}[w_1 \ldots w_n]$) and generates a set of candidate grammars on the basis of possible merges between any two minimal fragments in the grammar. Unlike in Stolcke and Omohundro's (1994) work, our merging operations do not include the original *Merge* operation, as our rules are categoryless, but only a variant of their *Chunk* operation, which creates novel minimal fragments or rules out of existing ones. We propose, different from Stolcke and Omohundro, that the search for optimal minimal fragments be guided by *pairwise* comparisons between known minimal fragments. These pairwise comparisons present us with structural analogies, which can be used to create novel minimal fragments on the basis of the analogically mappings. This process makes the grammar more compact, as shared structure in the minimal fragments is no longer redundantly stored in each of the trees, but in a new tree. On the other hand, it makes the probability of generating the observed combinations of subtrees less likely.

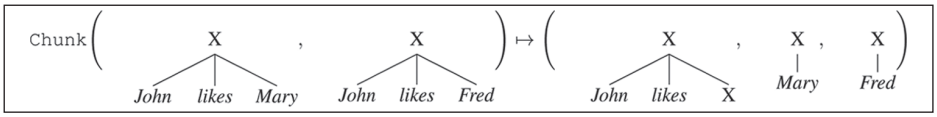

Figure 16. A Peripheral Chunk operation on *I laughed* and *So I laughed*.

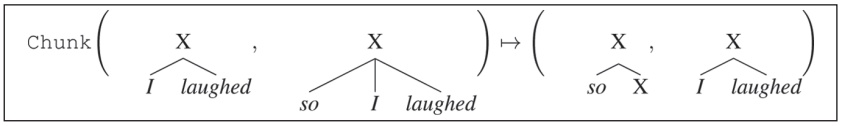

Figure 17. A Slot-filler Chunk operation on *John likes Mary* and *John likes Fred*.

More concretely, the model looks for compression of the grammar by optimally aligning two strings of leaf nodes of a minimal fragment by means of the backtrace of the Wagner–Fisher algorithm for finding the minimum edit distance between those two strings (Wagner & Fisher, 1974). This produces a set of identical leaf nodes and a set of non-identical ones. In two cases this mapping can lead to the creation of a novel minimal fragment.

1. One of the strings of leaf nodes is a proper subset of the other string of leaf nodes (Peripheral Chunk).
2. Each contiguous string of non-identical leaf nodes in each minimal fragment occurs in the same context as one contiguous string of non-identical leaf nodes in the other minimal fragment. A context can be a leaf node that is identical in both minimal fragments or the beginning or end of the string of leaf nodes (Slot-filler Chunk).

For *John likes Mary* and *John likes Fred*, this means that we find a Slot-filler Chunk: The differences *Mary* and *Fred* occur in the same context in both minimal fragments: between the word *likes* and the end of the string.

In principle, many other restrictions and other chunking operations could be posited, and future research will have to point out the effect of each. These two were selected to reflect a simple manner of analogical reasoning over minimal fragments. In case of a Peripheral Chunk, the difference between the fragments gives us the only novel fragment, as the overlap coincides with one of the two strings of leaf nodes (see Figure 16 for an example). In the Slot-filler Chunk, the mapped elements, with substitution sites at the positions of the differences form the leaf nodes of a new minimal fragment. Similarly, for all contiguous strings of differing leaf nodes, new minimal fragments containing those differing leaf nodes are added to the grammar as well (see Figure 17 for an example).

For each operation, we create a new candidate grammar of minimal fragments, differing slightly from the current grammar. As we apply the chunking operation to all pairs of minimal fragments, the procedure generates a set of candidate grammars, one for each legal chunking operation. Out of these, the candidate grammar that optimizes the compactness of the grammar and the fit with the data is selected to be the new grammar for the next round. This process of finding the candidate merges and selecting the optimal one is repeated until no better model can be found (with some tokens of lookahead to avoid getting stuck in local optima). The more compact the grammar and the data can be described, the less redundant the model is; information that was previously encoded

twice, is now encoded only once with the resulting rules from a chunking operation without too much loss in the description of the data.

The optimization of compactness and fit can be phrased in terms of encoding the information in the grammar and in the data as described by the grammar. These can be given by the *Minimum Description Length* principle (Rissanen, 1989). This principle states that the optimal model is the one that minimizes the sum of the length of the optimally compact binary encoding of the grammar given some encoding schema (the Grammar Description Length, *GDL*) and the length of the optimally compact binary encoding of the data given the grammar (the Data Description Length, *DDL*). These two terms are used because they trade-off the virtues of having a simple grammar and having a grammar that fits the data well:

$$DL = GDL + DDL$$

The *GDL* is the sum of the Description Lengths of all of the grammar's fragments. The Description Length of a minimal fragment, then, is the amount of bits needed to encode that rule.[5]

The *DDL* term makes use of the connection between information theory and probability theory. When we want to encode the data in an optimally compact manner, the fragments that will have to be often encoded should have a low cost of encoding. The optimal cost for each of the fragments used in the data points should thus be inversely proportional to its probability of occurrence. The description length of a single parse tree over a data point d, or $DL(d)$, is given by the logarithmic transformation of the joint probability of the minimal fragments $mf_1 \ldots mf_n$ that make up that parse tree, where the probability of a minimal fragment is given by its relative frequency among minimal fragments with an identical root node (i.e. all minimal fragments, as all are headed by the label X). The *DDL*, then, is the sum of the description lengths of all parse trees over all data points in the training data D:

$$DL(d) = -\log\left(\prod_{i=1}^{n} P(mf_i)\right)$$

$$DDL = \sum_{d \in D} DL(d)$$

The result of this optimization process is a set of minimal fragments out of which all training items are built up. Every round, the analyses of the training items are updated by replacing chunked fragments by the representations replacing them. After this grammar induction step, we can take all possible multi-fragment subtrees over our induced parse trees and use these as the productive units of the grammar. We will call this the extraction step. The expected effect of having an induction and an extraction step is that there is a reduction in the number of subtrees used, which does not hinder the performance, as the unsegmented chunks are unlikely to contain structure that is beneficial for parsing novel utterances. Effectively, this model, like other Bayesian approaches to structure learning (O'Donnell, 2011; Perfors et al., 2011; Post & Gildea, this issue), implements a cognitive version of Occam's razor by stating that the model that has as few and as simple 'entities' on the one hand and as good a coverage of the data as possible is the optimal model.

It might have struck the reader that the proposed induction algorithm operates by minimizing redundancy, whereas in Section 1 the authors proposed redundancy as a design feature of the cognitive system of a language user. There are two reasons why the resulting induced grammars still display redundancy. Firstly, the chunking operations take place locally. This means that it might be the case that *the boy* and *the girl* are chunked into *the X*, but a third item *the bathroom* is not chunked, possibly because of a higher likelihood of parsing *the bathroom* as one whole. Suppose

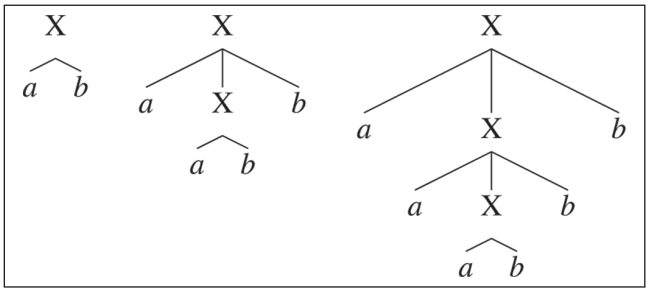

Figure 18. Optimal analyses of the strings *ab, aabb* and *aaabbb*.

now that *the X* is learned, and the model knows the word *bathroom* in some other way, and it still has *the bathroom* as a whole. In this case, the model displays redundancies, whereas the induced structure is optimal. Secondly, redundancy is introduced by extracting all subtrees from the induced parse trees. This is the kind of redundancy that DOP advocates and that has led to a good parsing performance as well as linguistically interesting findings. The redundancy-reducing optimization proposed is therefore an effect that is *local* rather than across the board and that applies to *finding the minimal bits* rather than the subtrees derived from the parse trees given by those minimal bits.

This model is by no means the first to explore data-driven restrictions on the hypothesis space of possible fragments. The issue that only some subtrees are useful in understanding novel linguistic material and means of finding those subtrees are discussed in what could be called 'best-subtree' approaches, such as Zuidema (2007), Cohn et al. (2009), Post and Gildea (2009, this issue) and O'Donnell (2011). The models described in those publications all start from analysed training items (but see Cohn, Blunsom, & Goldwater, 2010) and start their search process with grammars with small fragments (i.e. close to a probabilistic context-free grammar (PCFG) model that constitutes the base distribution). The bundle of properties that sets BMM-DOP apart from these approaches is that it unsupervisedly induces the bracketing structure instead of looking for substitution sites in a given parse tree, that it starts with large constructions and gradually moves towards smaller fragments, and that it does so by explicitly using analogies between different fragments to limit the hypothesis space.

3.2 First experiments

To show how the model arrives at linguistically interesting structures, we experimented with synthetic data as well as child-directed speech.

3.2.1 Synthetic data. A hallmark of context-free grammars is that they can recursively generate centre-embedded structures. The model presented is capable of learning rules that allow one to do so from a finite set of strings. Given that the model is presented with the strings *ab, aabb* and *aaabbb*, can the model learn that only strings of the form $a^n b^n$, with $n > 0$ and $n \in N$, is a legal production of this language?

The grammar induced from the three strings indeed shows the desired kind of centre embedding: performing a Peripheral Chunk on $_{X}[a\ b]$ and $_{X}[a\ a\ b\ b]$ (the former being a proper substring of the latter) yields $_{X}[a\ b]$ and $_{X}[a\ X\ b]$ as new rules, which amounts to a reduction of the Description Length (although the *DDL* increases, the *GDL* decreases more).[6] Eventually, the optimal

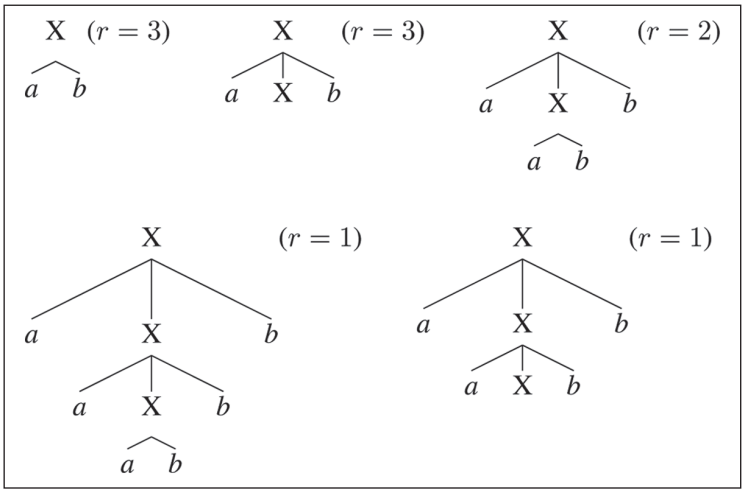

Figure 19. All subtree types from the analyses in Figure 18.

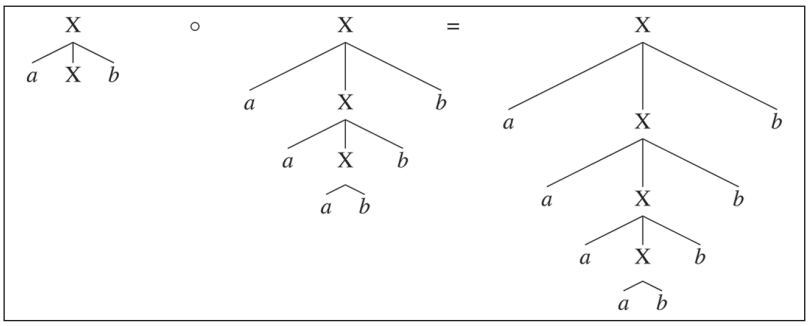

Figure 20. The most probable derivation of *aaaabbbb* given the subtrees in Figure 19.

grammar consists of the minimal fragments $_{X}[a\ b]$ and $_{X}[a\ X\ b]$.[7] Now, the resulting analyses of the training strings look as follows (Figure 18), with the set of all subtrees in Figure 19.

Given this bag of subtrees, the model is able to parse all and only those strings conforming to the format a^nb^n, for any $n > 0$. Interestingly, the all-subtrees assumption does present us with a cognitively interesting derivation process for an a^nb^n string with n being one larger than the largest observed n (e.g. *aaaabbbb* for our grammar). The most probable derivation for *aaaabbbb*, given a relative frequency estimator, is given in Figure 20, its probability being $3/10 \times 1/10 = 3/100$.

What is interesting is that this derivation consists of the depth-one subtree with a non-terminal node in the middle and a full, lexical fragment. Parsing a^nb^n strings for larger values of n thus amounts in this model to taking the largest possible, fully lexical chunks from memory and combining them with the smallest possible abstract chunk, basically making the number of embedding operations in the derivation only one instead of n–1 for an a^nb^n string analysed with only the minimal fragments $_{X}[a\ b]$ and $_{X}[a\ X\ b]$. This result suggests, in line with the principle of heterogeneity, that cases that seem complex (i.e. the number of centre embeddings used to generate an a^nb^n string using only depth-one subtrees) can sometimes be solved in a cognitively

simpler manner (i.e. the number of centre embeddings used to generate an $a^n b^n$ string using all subtrees, as in Figure 19).

A strong reason as to why the Model Merging-based approach arrives at the desired centre-embedding grammar is its chunking operation, which deterministically disables it to derive 'erroneous' grammars. U-DOP, not having these constraints, does indeed assign preferred analyses of $a^n b^n$ strings that are not centre-embedded.[8] It seems that the model thus has an explanatory advantage over U-DOP, but it remains to be seen if the model would be able to converge on the centre-embedding grammar given more flexible chunking operations that could lead the model astray.

3.2.2 Child-directed speech. If a child is to learn that the ambient language it is attending to is a hierarchical and productive system, it should be able to gather the evidence for this concept from the input it receives. Using the push-n-pull algorithm (cf., Zuidema, 2007) on the tree conversions of the dependency structure analyses of the Eve corpus (Brown, 1973), Borensztajn, Zuidema, and Bod (2009) showed how the most likely grammar to generate such data is more abstract at consecutive time points in development. Remaining questions are how such a grammar becomes more abstract and what influence the increased exposure to the ambient language has. The Model Merging approach provides us with a direction to look at for such an answer: the child finds more means of reducing the unwarranted redundancy in encoding the grammar and data as it processes more exemplars that are produced by a caregiver. As the reduction of redundancy amounts to more abstract grammars, there is effectively a causal connection between a greater amount of, and a more diverse nature of, the data and a more abstract grammar.

We test this prediction by looking at the analyses of the first 100 training items from the Eve corpus that came out after training on 100, 500, 1000 and 2000 items. This means that we performed only the parse tree induction step discussed in Section 3.1 and did not yet extract an all-subtrees grammar from the data in order to parse unseen data or reparse seen data, but simply explored the kind of parse trees that would be induced given the four different batches of data. We did so by looking at the average number of minimal fragments that compose an analysis. The higher the number of minimal fragments in the analysis of a training item, the more abstract and expressive the grammar is, as the space of generating unseen utterances increases. We also looked at the maximal depth of the analyses. The maximal depth is the longest path, counting the number of nodes crossed, from the top node to a terminal symbol. Here, the longer the paths are, the more embedded and hierarchical the syntax becomes. A grammar might use more rules to analyse an exemplar, but if these are used alongside each other instead of embedded in each other, the analyses remain relatively flat.

Let us first look at some examples. Figure 21 gives the analysis of three exemplars for the four different batches.

What does this tell us? As we can see, the overall abstraction and hierarchy increase over the four batches. In the first case, the learner notices that *can not* forms a constituent of some sort, but regards it as a variable or filler in batch II, and, seemingly more correctly, as a part of a frame in batches III and IV. Similarly, in the second example, the first analysis is almost flat, with the exception of the recognized word *bottle*. In batch II, hierarchy is discovered in this sentence, but *dollie* and the possessive pronoun *her* are erroneously taken to be a constituent, which is corrected after having processed 500 more exemplars in batch III. Furthermore, starting from batch II we see that there is some sort of an imperative verb-island construction $_{\text{X}}$[*give* X X], or *give* with two arguments. Unfortunately, the development is not always towards a more intuitively likely analysis: in example three the analysis based on batch II seems the most plausible: *a* and *baby* form a

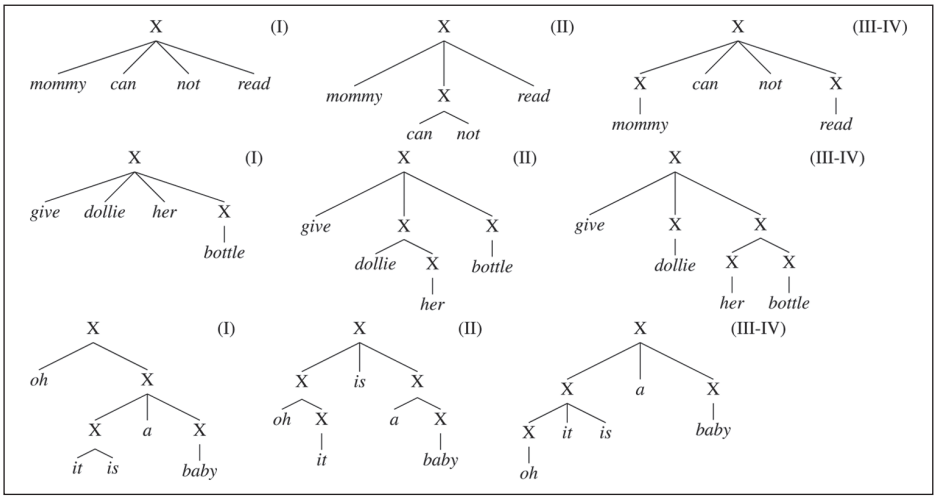

Figure 21. The analysis of three exemplars given the optimal minimal fragments after the four different batches (I = 100 items, II = 500, III = 1000, IV = 2000).

Table 1. The average number of minimal fragments and the average maximal depth in the analyses of the first 100 training items, given batches of 100, 500, 1000 and 2000 training items.

n training items	*n* minimal fragments	Maximal depth
100	2.25	1.90
500	2.80	2.18
1000	3.03	2.33
2000	3.53	2.55

constituent and there is something like a copular-construction *X is X*, which both lack in the other analyses. An important part of the further exploration of this model is to see to what extent the analyses conform more to some hand-annotated standard.

Turning to a quantitative analysis, the prediction of increasing hierarchy and abstraction was borne out by the data, as can be seen in Table 1.

As we can see, an analysis of one of the first 100 utterances by the mother has on average more than one minimal fragment more after processing 2000 exemplars than after processing only those 100. Note that the upper bound, for a fully abstract grammar consisting of the rules $X \rightarrow XX$ and $X \rightarrow t$ for any terminal symbol t, is an average of 7.94 minimal fragments per analysis. This means that the optimal grammar, after having seen 2000 training items, is not as abstract as it could in principle be. It remains to be seen how close to this upper bound a grammar trained on an abundance of non-child-directed speech would be.

As for the average depth per exemplar, we can see that this figure increases monotonously as well, meaning that as the learner processes more exemplars, the analyses become more hierarchical. Both findings taken together, we can see that Borensztajn et al.'s (2009) finding on the production side can be mimicked on the comprehension side by a learning model that carves up the analyses of processed experiences according to analogies and Description Length biases.

4 Conclusion

In this paper we presented three design principles of language, redundancy featuring among them. We gave an overview of a class of models that exploits redundancy both in language parsing and language learning (DOP and U-DOP). Although the idea of some form of redundant storage has become part and parcel of parsing technologies and usage-based linguistic approaches alike, the question *how much* of it is cognitively realistic and/or computationally optimally efficient is an open one. A segmentation strategy, such as Model Merging, paired with an all-subtrees approach could reduce the number of rules needed to achieve an optimal performance, thus making the parser and learner more efficient. At the same time, starting from unsegmented wholes comes closer to the acquisitional situation of a language learner, and thus adds to the cognitive plausibility of the model.

Funding

We gratefully acknowledge funding by the Netherlands Organization for Scientific Research (NWO): BB was funded through a Promoties in de Geesteswetenschappen-grant "Constructions Emerging" (322.70.001), RB through a Vici-grant "Integrating Cognition" (277.70.006), and WZ through a Veni-grant "Discovering Grammar" (639.021.612). We also thank 3 anonymous reviewers for useful comments.

Notes

1. As we will be talking only about redundancy of representation (the storage of multiple elements covering the same material) and not about signal redundancy, we will henceforth refer to the former concept with the term *redundancy*, lest our terminology be redundant. Note that representational redundancy also implies *derivational* redundancy: if there is redundant storage, there can be expected to be multiple ways of arriving at the same analysis of a datum given these representations.
2. This restriction can be easily relaxed. Yet U-DOP does not learn argument structure (cf. Alishahi & Stevenson, 2008) or any other semantic information.
3. In Bod (2009), a further extension of U-DOP is proposed that takes into account the shortest derivation as well.
4. We are implicitly assuming a DOP model that computes the most probable sentence given a certain meaning to be conveyed, such as in Bonnema, Bod, and Scha (1997) and Bod (1998).
5. More precisely, the Description Length of a minimal fragment is given by the number of bits needed to declare the fragment (1 bit) in order to demarcate it from other fragments, encode for each leaf node if it is a terminal or a non-terminal leaf node (1 bit for each leaf node) and, in case it is a terminal leaf node, encode what category ($-log_2$|terminal category| bits per terminal leaf node) and finally end the fragment (1 bit). Although more efficient coding schemas can be developed (cf. Petasis, Paliouras, Karkaletsis, Halatsis, & Spyropoulos, 2004), as a first approach we employ this simple schema.
6. Note that given the original Merge and Chunk operators of Stolcke and Omohundro (1994), an $a^n b^n$ grammar could also be induced.
7. The cost for declaring a terminal symbol in the grammar is $\log_2(2)$, as the terminal symbols consists of$\{a,b\}$. Following the encoding schema's in Section 3.1, for $X \rightarrow aXb$ and $X \rightarrow ab$ the $GDL = (1 + 2(1 + \log_2(2)) + 1) + (1 + 2(1 + \log_2(2))) = 12$, whereas the DDL is $-\log_2(3/6 \times 3/6 \times 3/6 \times 3/6 \times 3/6 \times 3/6) = -6\log_2(3/6) = 6$, given that $r(X \rightarrow aXb) = 3$ and $r(X \rightarrow ab) = 3$ for our three training items. We leave the calculations of the DLs of the suboptimal grammars to the readers.
8. Gideon Borensztajn, personal communication.

References

Abney, S. (1996). Statistical methods and linguistics. In J. Klavans & P. Resnik (Eds.), *The balancing act: Combining symbolic and statistical approaches to language* (pp. 1–26). Cambridge, MA: The MIT Press.

Alishahi, A., & Stevenson, S. (2008). A computational model for early argument structure acquisition. *Cognitive Science*, *32*, 789–834.

Arnon, I., & Cohen-Priva (this issue). More than words: The effect of multi-word frequency and constituency on phonetic duration. *Language and Speech*

Arnon, I., & Snider, N. (2010). More than words: Frequency effects for multi-word phrases. *Journal of Memory and Language*, *62*, 67–82.

Bannard, C., & Matthews, D. (2008). Stored word sequences in language learning: The effect of familiarity on children's repetition of four-word combinations. *Psychological Science*, *19*, 241–248.

Bansal, M., & Klein, D. (2010). Simple, accurate parsing with an all-fragments grammar. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 1098–1107.

Bod, R. (1992). A computational model of language performance: data-oriented parsing. *Proceedings of the fifteenth International Conference on Computational Linguistics*, pp. 855–859.

Bod, R. (1998). *Beyond grammar: An experienced-based theory of language*. Stanford, CA: CSLI Publications.

Bod, R. (2001). Sentence memory: Storage vs. computation of frequent sentences. *Proceedings of the CUNY Conference on Sentence Processing 2001*, Philadelphia, PA.

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 of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 865–872.

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

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

Bod, R., Hay, J., & Jannedy, S. (Eds.) (2003). *Probabilistic linguistics*. Cambridge, MA: The MIT Press.

Bod, R., Scha, R., & Sima'an, K. (Eds.) (2003). *Data-oriented parsing*. Stanford, CA: CSLI Publications.

Bod, R., & Smets, M. (2012). Empiricist solutions to nativist puzzles by means of unsupervised TSG. *Proceedings of the Workshop on Computational Models of Language Acquisition and Loss at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 10–18.

Bonnema, R., Bod, R., & Scha, R. (1997). A DOP model for semantic interpretation. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pp. 159–167.

Borensztajn, G., Zuidema, W., & Bod, R. (2009). Children's grammars grow more abstract with age. *Topics in Cognitive Science*, *1*, 175–188.

Bresnan, J. (2007). Is syntactic knowledge probabilistic? Experiments with the English dative alternation. In S. Featherston & W. Sternefeld (Eds.), *Roots: Linguistics in search of its evidential base* (pp. 77–96). Berlin, Germany: Mouton de Gruyter.

Brown, R. (1973). *A first language: The early stages*. London, UK: George Allen & Unwin Ltd.

Bybee, J. (2006). From usage to grammar: The mind's response to repetition. *Language*, *82*, 711–733.

Charniak, E., & Johnson, M. (2005) Coarse-to-fine n-best parsing and maxent discriminative reranking. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Chomsky, N. (1957). *Syntactic structures*. The Hague, The Netherlands: Mouton De Gruyter.

Clark, A., & Eyraud, R. (2006). Learning auxiliary fronting with grammatical inference. Proceedings of the Tenth Workshop on Natural Language Learning.

Cohn, T., Blunsom, P., & Goldwater, S. (2010). Inducing Tree-Substitution Grammars. *Journal of Machine Learning Research*, *11*, 3053–3096.

Cohn, T., Goldwater, S., & Blunsom, P. (2009). Inducing compact but accurate tree-substitution grammars. *Proceedings of Human Language Technologies: The 2009 Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 548–556.

Collins, M., & Duffy, N. (2001). Convolution kernels for natural language. *Advances in Neural Information Processing Systems*, *13*, 625–632.

Crain, S. (1991). Language acquisition in the absence of experience. *Behavorial and Brain Sciences*, *14*, 597–612.

Crain, S., & Nakayama, M. (1987). Structure dependence in grammar formation. *Language*, *63*, 522–543.

Culicover, P., & Jackendoff, R. (2005). *Simpler syntax*. Oxford, UK: Oxford University Press.
Frank, S., Bod, R., & Christiansen, M. (2012). How hierarchical is language use? *Proceedings of the Royal Society B*, *279*, 4522–4531.
Gahl, S., & Garnsey, S. (2004). Knowledge of grammar, knowledge of usage: Syntactic probabilities affect pronunciation variation. *Language*, *80*, 748–775.
Gahl, S., & Garnsey, S. (2006). Knowledge of grammar includes knowledge of syntactic probabilities. *Language*, *82*, 405–410.
Goldberg, A. (2006). *Constructions at work. The nature of generalization in language*. Oxford, UK: Oxford University Press.
Goodman, J. (2003). Efficient parsing of DOP with PCFG-reductions. In R. Bod, R. Scha, & K. Sima'an (Eds.), Data-Oriented Parsing (pp. 125–146). Stanford, CA: CSLI Publications.
Hopper, P., & Traugott, E. (2003). *Grammaticalization*. Cambridge, UK: Cambridge University Press.
Johnson, M., Griffiths, T., & Goldwater, S. (2007). Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. *Advances in Neural Information Processing Systems*, *19*, 641–648.
Kam, X., Stoyneshka, L., Tornyova, L., Fodor, J., & Sakas, W. (2008). Bigrams and the richness of the stimulus. *Cognitive Science*, *32*, 771–787.
Kaplan, R. (1996). A probabilistic approach to lexical-functional analysis. *Proceedings of the 1996 LFG Conference and Workshops*.
Lewis, J., & Elman, J. (2001). Learnability and the statistical structure of language: Poverty of stimulus arguments revisited. *Proceedings of 26th annual Boston University Conference on Language Development*, pp. 359–370.
MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk*. Mawah, NJ: Lawrence Erlbaum Associates.
MacWhinney, B. (2005). Item-based constructions and the logical problem. *Proceedings of the Second Workshop on Psychocomputational Models of Human Language Acquisition*, Ann Arbor, MI.
Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: The MIT Press.
Marr, D. (1982). *Vision. A computational approach*. San Francisco, CA: Freeman & Co.
Newmeyer, F. (2006). On Gahl and Garnsey on grammar and usage. *Language*, *82*, 399–404.
O'Donnell, T. (2011). *Productivity and reuse in language*. PhD thesis, Harvard University.
Perfors, A., Tenenbaum, J. B., & Regier, T. (2011). The learnability of abstract syntactic principles. *Cognition*, *118*, 306–338.
Petasis, G., Paliouras, G., Karkaletsis, V., Halatsis, C., & Spyropoulos, C. (2004). E-grids: Computationally efficient grammatical inference from positive examples. *Grammars*, *7*, 69–110.
Post, M., & Gildea, D. (2009). Bayesian learning of a tree substitution grammar. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*.
Post, M., & Gildea, D. (this issue). Bayesian tree substitution grammars as a usage-based approach. *Language and Speech*
Reali, F., & Christiansen, M. (2005). Uncovering the richness of the stimulus: structure dependence and indirect statistical evidence. *Cognitive Science*, *29*, 1007–1028.
Rissanen, J. (1989). *Stochastic complexity in statistical inquiry*. Singapore: World Scientific.
Sangati, F., & Zuidema, W. (2011). Accurate parsing with compact tree-substitution grammars: double-DOP. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 84–95.
Scha, R. (1990). Taaltheorie en taaltechnologie; Competence en performance. In Q. de Kort & G. Leerdam (Eds.), *Computertoepassingen in de Neerlandistiek* (pp. 7–22). Almere, The Netherlands: Landelijke Vereniging van Neerlandici.
Sima'an, K. (1996). Computational complexity of probabilistic disambiguation by means of tree grammars. *Proceedings of the 16th International Conference on Computational Linguistics*, pp. 1175–1180.

Spivey, M. J., & Tanenhaus, M. K. (1998). Syntactic ambiguity resolution in discourse: Modeling the effects of referential context and lexical frequency. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *24*, 1521–1543.

Stolcke, A., & Omohundro, S. (1994). Inducing probabilistic grammars by Bayesian model merging. In R. C. Carrasco, & J. Oncina (Eds.), *Grammatical inference and applications* (pp. 106–118). Berlin, Germany: Springer.

Tremblay, A., & Baayen, R. H. (2010). Holistic processing of regular four-word sequences: A behavioral and ERP study of the effects of structure, frequency, and probability on immediate free recall. In D. Wood (Ed.), *Perspectives on formulaic language: Acquisition and communication* (pp. 151–173). London, UK: The Continuum International Publishing Group.

Verhagen, A. (2005). *Constructions of intersubjectivity: Discourse, syntax and cognition*. Oxford, UK: Oxford University Press.

Wagner, R. A., & Fisher, M. J. (1974). The string-to-string correction problem. *Journal of the Association for Computing Machinery*, *21*, 168–173.

Zuidema, W. (2006). Theoretical evaluation of estimation methods for Data-Oriented Parsing. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 183–186 (corrigendum at staff.science.uva.nl/~jzuidema).

Zuidema, W. (2007). Parsimonious data-oriented parsing. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 551–560.