

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 be 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 Childe database, which indicates that children learn discontinuous 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 Childe 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.

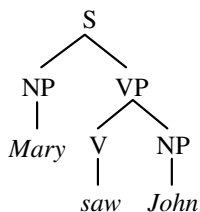


Figure 1. Phrase structure tree for *Mary saw John*

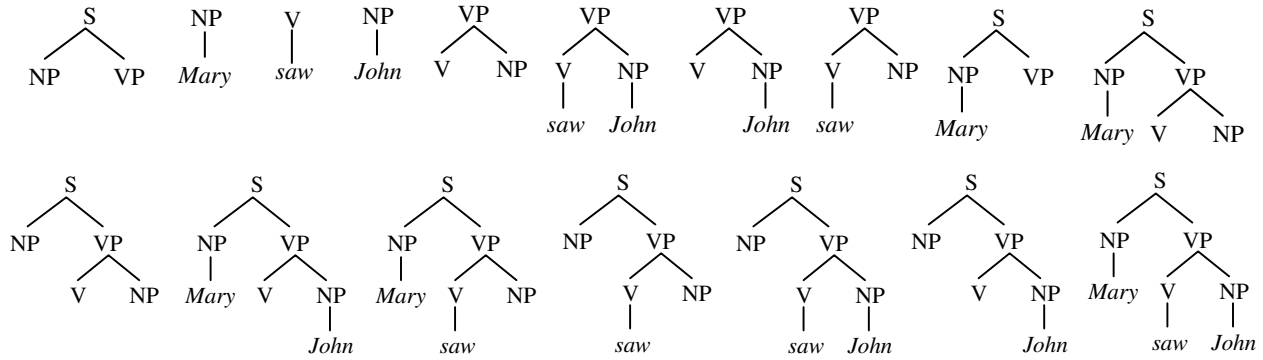


Figure 2. Subtrees from the tree in Figure 1

Thus the top-leftmost subtree in Figure 2 is equivalent to the traditional context-free rewrite rule $S \rightarrow NP VP$, 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.

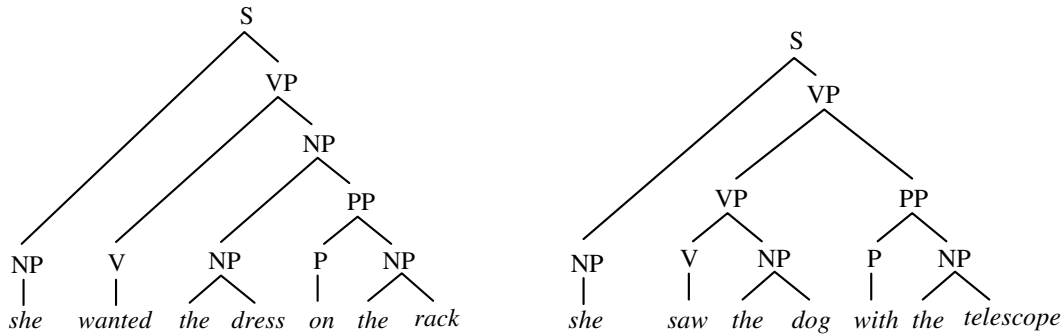


Figure 3. An extremely small corpus of two phrase-structure trees

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 \circ , 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 leftmost nonterminal leaf node of the first subtree provided that their categories match.

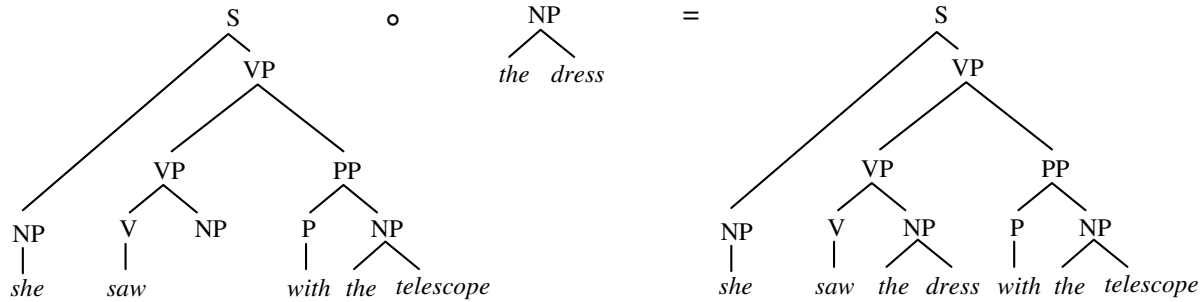


Figure 4. Analyzing a new sentence by combining subtrees from Figure 3

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.

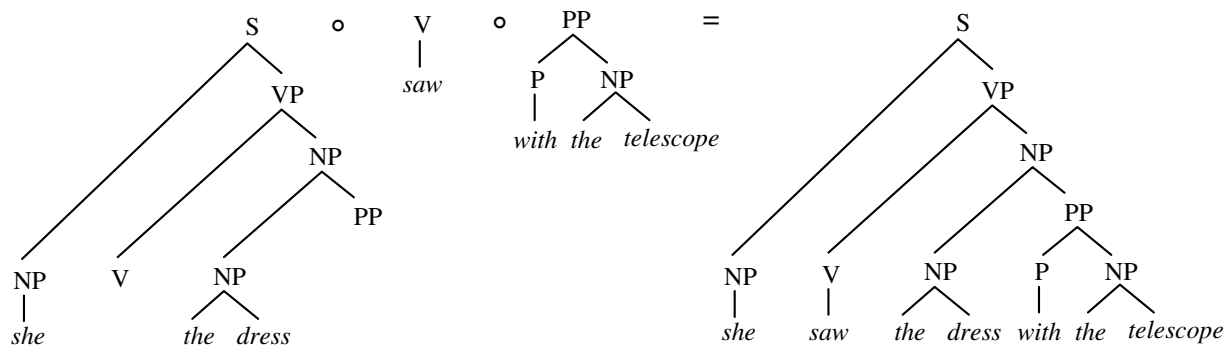


Figure 5. A different derivation for *She saw the dress with the telescope*

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, \dots, 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, \dots, t_k in T_1, T_2, \dots, T_n for which $t_1 \circ t_2 \circ \dots \circ t_k = T$, 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 $d = t_1 \circ \dots \circ t_k$ 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:

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

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 $t_1 \circ \dots \circ t_k$ is defined as the product of the probabilities of its subtrees t_i :

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

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 :

$$T_{best} = \operatorname{argmax}_{T_{sd}} P(T_{sd} | W)$$

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).

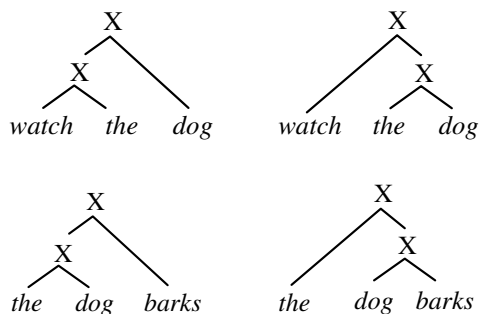


Figure 6. The unlabeled binary tree set for *watch the dog* and *the dog barks*

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.

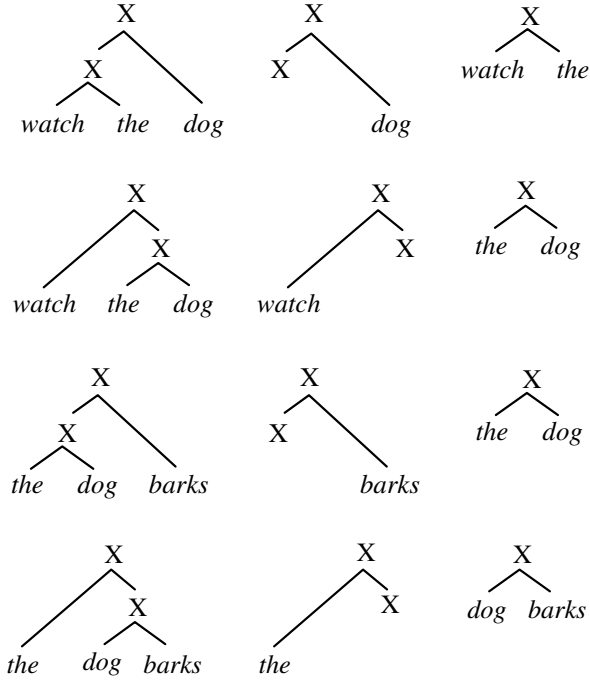


Figure 7. The subtree set for the binary trees in Figure 6.

Note that while most subtrees occur once, the subtree $[the\ dog]_X$ 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.

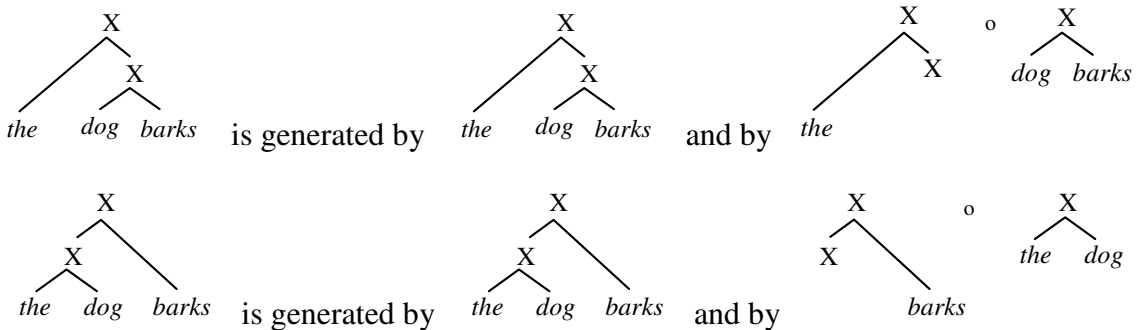


Figure 8. Parsing *the dog barks* from the subtrees in Figure 7

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.

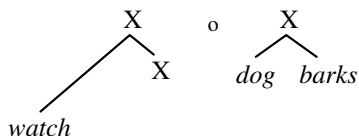


Figure 9. Unique shortest derivation for *watch dog barks* from the subtrees in Figure 7

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:

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

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 discontinuous yield *watch X dog*, which we will therefore refer to as a *discontinuous subtree*.

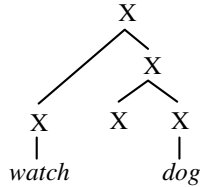


Figure 10. A discontinuous subtree

Discontinuous 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 discontinuous. 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 discontinuous constructions. For example, the construction *more ... than ...* in (1) may be represented by the subtree in Figure 11.

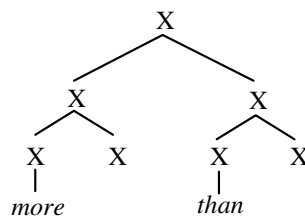


Figure 11. Discontinuous subtree for the construction *more...than...*

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 discontinuous, 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 discontinuous 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).

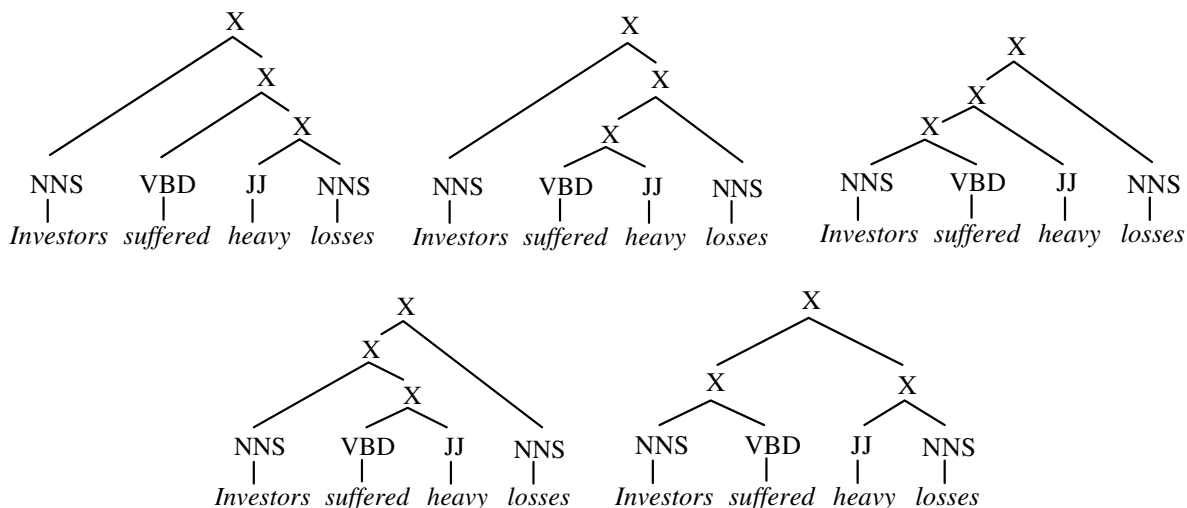


Figure 12. All binary trees for the WSJ sentence *Investors suffered heavy losses*

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 discontinuous 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 discontinuous 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 discontinuous, while the other two subtrees (b) and (c) are contiguous.

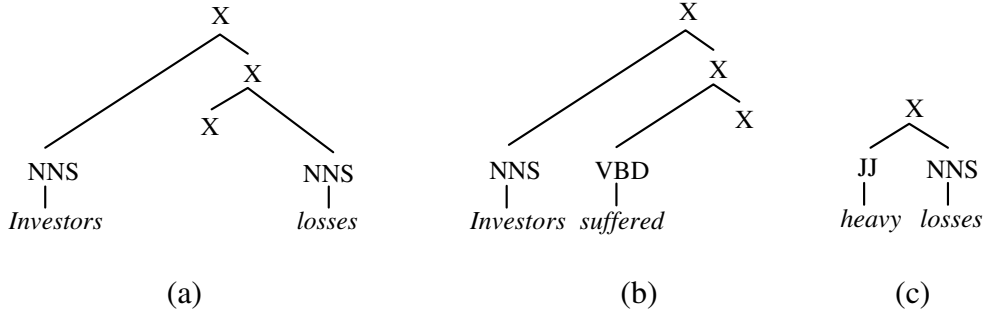


Figure 13. One discontinuous subtree and two contiguous subtrees from Figure 12

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-occurrences 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 constituent 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 constituent DT JJ. In other words: while constituents 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 Childe database

To test U-DOP on child language, we used the Eve corpus (Brown 1973) in the Childe 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.

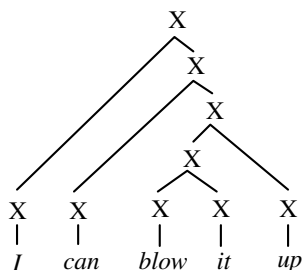


Figure 14. Binary tree for Childes sentence *I can blow it up* (Eve corpus)

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 Chiles 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 Chiles, 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, $p < 0.01$).

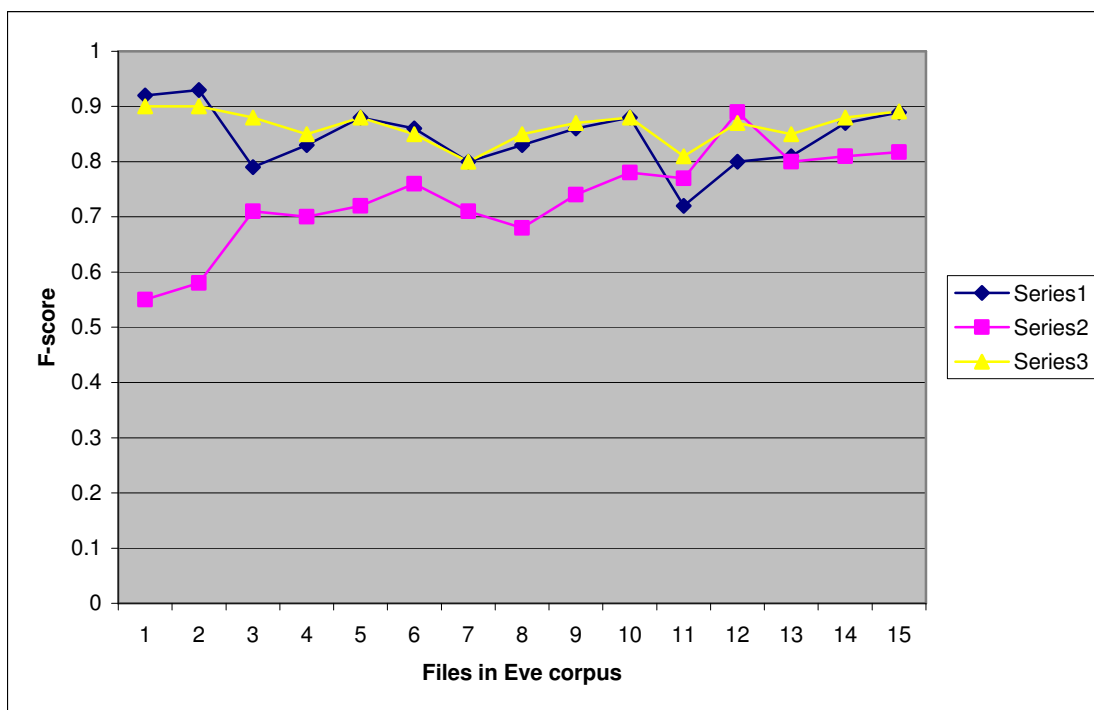
Table 4 shows that the use of all subtrees consistently outperforms the use of only contiguous subtrees. This is consonant with our results in the previous section. An additional experiment with 10-fold testing showed that the differences in f-score between full U-DOP and U-DOP without discontiguous 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 discontinuous 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 increase 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 discontinuous 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: $[[\textit{blow X}] \textit{up}]$ and $[\textit{blow} [\textit{X up}]]$ without breaking ties yet. Finally, in sentence 15, Eve produces *blow it* without *up*, which led U-DOP to assign the subtree $[[\textit{blow X}] \textit{up}]$ a higher frequency than $[\textit{blow} [\textit{X up}]]$. 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 discontinuous 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 constructions 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.

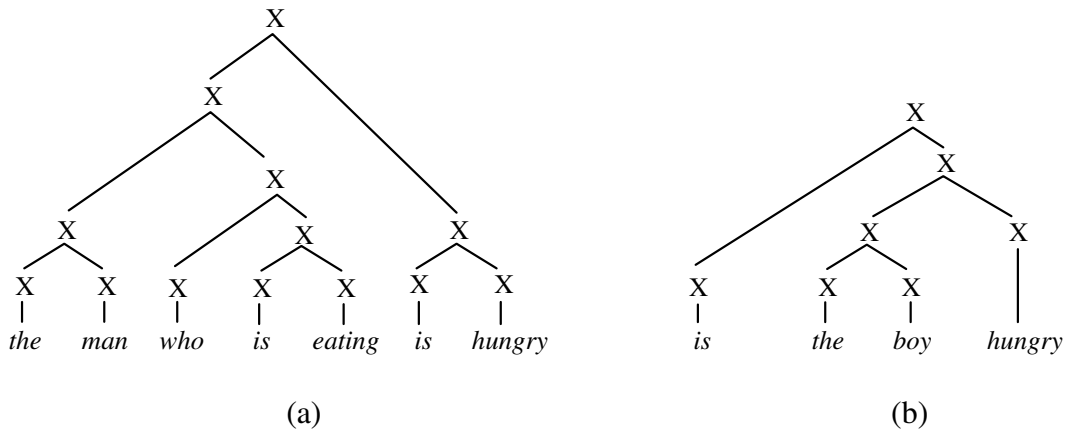


Figure 18. Tree structures for *the man who is eating is hungry* and *is the boy hungry?* learned by U-DOP from the sentences (10)-(15)

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 discontinuous)⁹.

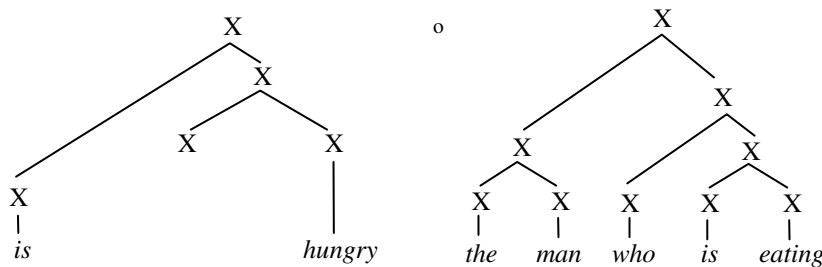


Figure 19. Producing the correct auxiliary fronting by combining two subtrees from Figure 18

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:

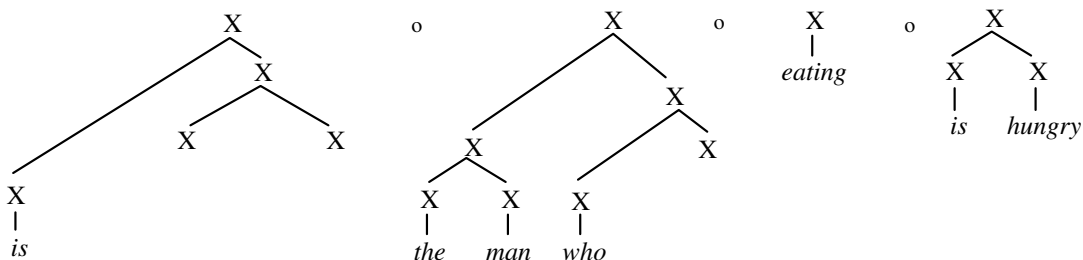


Figure 20. Producing the incorrect aux-fronting by combining four subtrees from Figure 18

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 Bonnama 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 $[is_x X]_x$ 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 auxiliary 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 auxiliary 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
<u>Other:</u>	
*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 <i>Is P who is QR?</i> 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 discontinuous 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 discontinuous 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 discontinuous 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 discontinuity 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 $C_1 \dots C_n$) and let the MPSP 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, discontinuous 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, Chomsky 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 $a = \sum_j a_j$. 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:

$$\begin{array}{ll}
 A_j \rightarrow BC & (1/a_j) & A \rightarrow BC & (1/a) \\
 A_j \rightarrow B_k C & (b_k/a_j) & A \rightarrow B_k C & (b_k/a) \\
 A_j \rightarrow BC_1 & (c_1/a_j) & A \rightarrow BC_1 & (c_1/a) \\
 A_j \rightarrow B_k C_1 & (b_k c_1/a_j) & A \rightarrow B_k C_1 & (b_k c_1/a)
 \end{array}$$

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 constrained.

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 weight, 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 children'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. *Behavioral 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.
- Gener, 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. Stoyaneshka, L. Torniyova, 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, Mahwah, 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.