LINGUISTIC CONSTRAINTS IN LFG-DOP

Doug Arnold and Evita Linardaki
University of Essex

Proceedings of the LFG07 Conference

Miriam Butt and Tracy Holloway King (Editors)

2007

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

LFG-DOP (Bod and Kaplan, 1998, 2003) provides an appealing answer to the question of how probabilistic methods can be incorporated into linguistic theory. However, despite its attractions, the standard model of LFG-DOP suffers from serious problems of overgeneration because (a) it is unable to define fragments of the right level of generality, and (b) it has no way of capturing the effect of anything except simple positive constraints. We show how the model can be extended to overcome these problems.

1 Introduction

The question of how probabilistic methods should be incorporated into linguistic theory is important from both a practical, grammar engineering, perspective, and from the perspective of 'pure' linguistic theory. From a practical point of view such techniques are essential if a system is to achieve a useful breadth of coverage and avoid being swamped by structural ambiguity in realistic situations. From a theoretical point of view they are necessary as a response to the influence of probabilistic factors in human language behaviour (see e.g. Jurafsky, 2003, for a review).

Bod and Kaplan (1998, 2003) provide a very appealing and persuasive answer to this question in the form of LFG-DOP, where the linguistic representations of Lexical Functional Grammar (LFG) are combined with the probabilistic methods of Data Oriented Parsing (DOP). The result is a descriptively powerful, clear, and elegant fusion of linguistic theory and probability. However, it suffers from two serious problems, both related to generative capacity, which have the effect that the model overgenerates. This paper shows how these problems can be overcome.

The paper is structured as follows. Section 2 provides background, introducing the basic ideas of DOP. Section 3 describes the Bod and Kaplan (B&K) model, and introduces the first problem: the problem of defining DOP fragments with the right level of generality. Section 4 shows how this problem can be overcome. Section 5 describes the second problem (which arises because LFG-DOP fragments effectively encode only simple, positive, LFG constraints) and shows how it can be overcome. Section 6 discusses some issues and potential objections.

2 Tree-DOP

The central idea of DOP is that, rather than using a collection of rules, parsing and other processing tasks employ a database of *fragments* produced by decomposing a collection of normal linguistic representations (e.g. trees drawn from a

[†] We are grateful to the participants at LFG07 in Stanford, Ca, for insightful and stimulating discussion, in particular: Joan Bresnan, Aoife Cahill, Grzegorz Chrupała, Ron Kaplan, Jonas Kuhn, and Louisa Sadler.

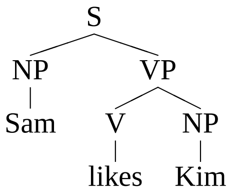


Figure 1: Treebank representation

treebank).[1] These fragments can be assigned probabilities (e.g. based on their relative frequency of appearance in the fragment database). Parsing a string involves, in effect, finding a collection of fragments which can be combined to derive it, i.e. provide a representation for it. These representations are assigned probabilities based on the probabilities of the fragments used. This general approach can, of course, be realized in many different ways, via different choices of basic representation, different decomposition operations, etc. So, standardly, specifying a DOP model involves instantiating four parameters: (i) representational basis; (ii) decomposition operations; (iii) composition operation(s); and (iv) probability model.

Specified in this way, Tree-DOP, the simplest DOP model, involves:

(i) a treebank of context free trees, such as Figure 1;
(ii) two decomposition operations: $Root$ and $Frontier$;
(iii) a single composition operation: *Leftmost Substitution*;
(iv) a probability model based on relative frequency.

Fragments are produced from representations such as Figure 1 by two decomposition operations: $Root$ and $Frontier$:

(i) $Root$ selects any node n and makes it the root of a new tree, erasing all other nodes apart from those dominated by n.
(ii) $Frontier$ chooses a set of nodes (other than the root) and erases all subtrees dominated by these nodes.

Intuitively, $Root$ extracts a complete constituent to produce a fragment with a new root. For example, the fragments in Figure 2 can be produced from the tree in Figure 1 by (possibly trivial) application of $Root$. $Frontier$ deletes part of a fragment to produce an 'incomplete' fragment — a fragment with a new frontier containing 'open slots' (i.e. terminal nodes labeled with a non-terminal category), as in Figure 3.

Leftmost Substitution involves substituting a fragment for the leftmost open slot. Figure 4 exemplifies one of the several ways in which a representation of *Kim likes Sam* can be derived.

[1] Standard references on DOP include, for example, Bod and Scha (1997), Bod (1998), and the papers in Bod et al. (2003). All of these contain presentations of Tree-DOP.

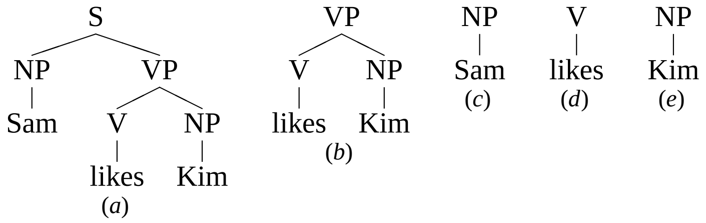


Figure 2: Fragments produced by the *Root* operation

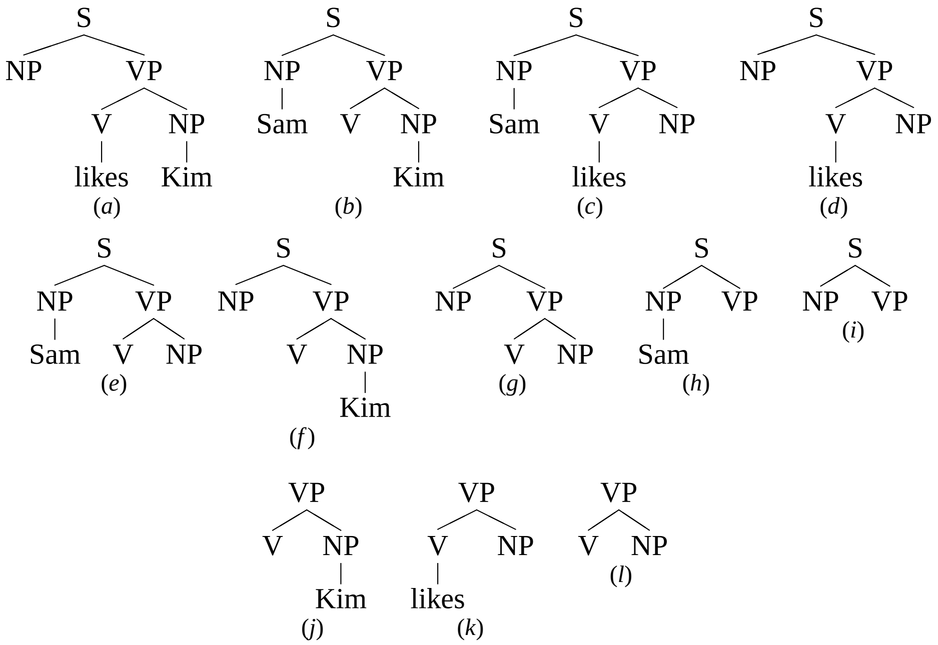


Figure 3: Fragments produced by the *Frontier* operation

The following define a very simple probability model for this version of DOP.[2]

(1)
$$P(f_i) = \frac{|f_i|}{\displaystyle\sum_{root(f)=root(f_i)} |f|}$$

(2)
$$P(d) = \prod_{i=1}^{n} P(f_i)$$

[2] Simple, and one should add, inadequate. This model is based on relative frequency estimation, which has been shown to be biased and inconsistent (Johnson, 2002). A number of alternatives have been proposed, e.g. assuming a uniform derivation distribution (Bonnema et al., 1999), backing-off (Sima'an and Buratto, 2003), and held-out estimation (Zollmann, 2004). Nothing in what follows depends on the choice of probability model, however.

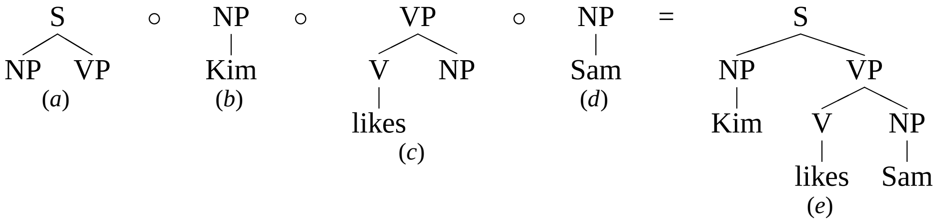


Figure 4: Fragment composition

(3) $$P(R) = \sum_{j=1}^{m} P(d_j)$$

Equation (1) says that the probability associated with a fragment f_i is the ratio of the number of times it occurs compared to the number of times fragments with the same root category occur. (2) says that the probability of a particular derivation d is the product of the probabilities of the fragments used in deriving it. (3) says that the probability associated with a representation (tree) is to be found by summing over the probabilities of its derivations.

Apart from its obvious simplicity, this version of DOP has numerous attractions. However, from a linguistic point of view it suffers from the limitations of the underlying linguistic theory (context-free phrase structure grammar), and for this reason does not provide a satisfactory answer to the question of how probabilistic and linguistic methods should be combined. A much better answer emerges if DOP techniques are combined with a richer linguistic theory, such as LFG.[3]

3 LFG-DOP

The idea of combining DOP techniques with the linguistic framework of LFG was first proposed in Bod and Kaplan (1998) (see also Bod and Kaplan, 2003; Way, 1999; Bod, 2000b,a; Hearne and Sima'an, 2004; Finn et al., 2006; Bod, 2006). As one would expect given the framework, representations are triples $\langle c, \phi, f \rangle$, consisting of a c-structure, an f-structure, and a 'correspondence' function ϕ that relates them (see Figure 5).[4]

Decomposition again involves the $Root$ and $Frontier$ operations. As regards c-structure, these operations are defined precisely as in Tree-DOP. However, the operations must also take account of f-structure and the ϕ-links: (i) when a node is erased, all ϕ-links leaving from it are removed, and (ii) all f-structure units that are not ϕ-accessible from the remaining nodes are erased.[5] (iii) In addition, $Root$

[3] Attempts to adapt DOP for other grammatical formalisms, notably HPSG, include Neumann (2003), Linardaki (2006), and Arnold and Linardaki (2007).

[4] Discussion of the key ideas of LFG can be found in e.g. Bresnan (1982), Dalrymple et al. (1995), Bresnan (2001), and Dalrymple (2001).

[5] A piece of f-structure is ϕ-accessible from a node n if and only if it is ϕ-linked to n or contained within a the piece of f-structure that is ϕ-linked to n.

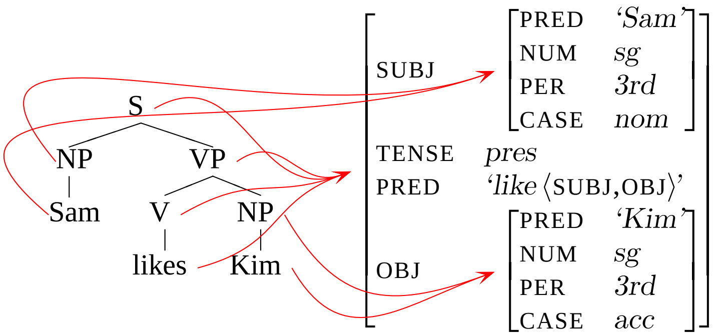


Figure 5: LFG-DOP Treebank representation.

deletes all semantic forms (PRED features) that are local to f-structures which are linked to erased nodes. (iv) *Frontier* also removes semantic forms from f-structures corresponding to erased nodes.

The intuition here is (a) to eliminate f-structure that is not associated with the c-structure that remains in a fragment, and (b) to keep everything else, except that a fragment should contain a PRED value if and only if the c-structure contains the corresponding word. Thus, from the representation in Figure 5, *Root* will produce (*inter alia*) fragments corresponding to the NPs *Sam* and *Kim* and the VP *likes Kim*, as in Figure 6. The cases of *Sam* and *Kim* are straightforward: all other nodes, and the associated ϕ-links have been removed; the only f-structures that are ϕ-accessible are the values of SUBJ and OBJ respectively, and these are what appear in the fragments. The case of the VP *likes Kim*, is slightly more complex: deleting the S and subject NP nodes does not affect ϕ-accessibility relations, because the S and VP nodes in Figure 5 are ϕ-linked to the same f-structure. However, deleting the subject NP removes the PRED feature the SUBJ value, as required by (iii). Notice that nothing else is removed: in particular, notice that person-number information about the subject NP remains.

Applying *Frontier* to Figure 6 (*c*) to delete *Kim* will produce a fragment corresponding to *likes NP*, as in Figure 7. Again, ϕ-accessibility is not affected, so the only effect on the f-structure is the removal of the PRED feature associated with *Kim*, as required by (iv).

The composition operation will not be very important in what follows. For the purpose at hand it can be just the same as that of Tree-DOP, with two provisos. First, we must ensure that substitution of a fragment at a node preserves ϕ-links and also unifies the corresponding f-structures. Second, we require the f-structure of any final representation we produce to satisfy a number of additional well-formedness conditions, specifically *uniqueness*, *completeness* and *coherence*, in the normal LFG sense (e.g. Dalrymple, 2001, pp35-39). Similarly, for the purpose of this discussion we can assume the probability model is the same as used in Tree-DOP. [6]

[6] In fact, a small extension of the probability model is needed. *Completeness* cannot be checked in

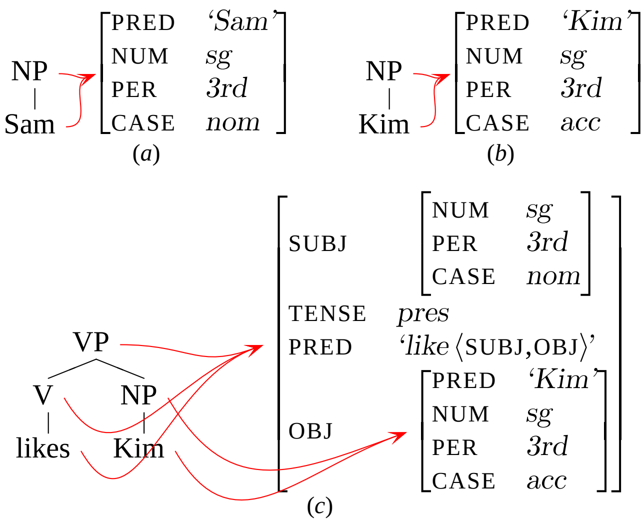


Figure 6: LFG-DOP *Root* fragments

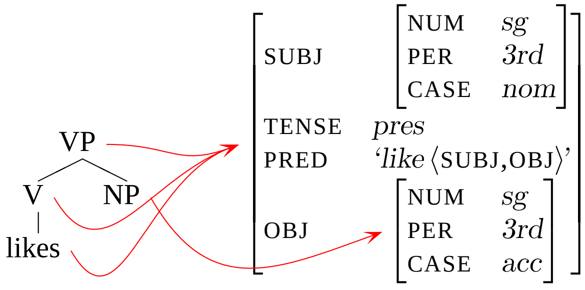


Figure 7: An LFG-DOP *Frontier* fragment

What is of central concern here is that the fragments produced by *Root* and *Frontier* are highly *undergeneral* (overspecific). In particular, the fragment for *Sam* is *nom*, the fragment for *Kim* is *acc*, and in the fragment for *likes NP* the direct object NP is third person and singular.

This will lead to under-generation (under-recognition). For example, it will not be possible to use the *Root* fragments for *Sam* and *Kim* in Figure 6 in analyzing a sentence like (4) where *Kim* appears as a subject, and *Sam* as an object, because they have the wrong case marking. Similarly, it will not be possible to use the *Frontier* fragment in Figure 7 to analyze (5), since it requires the OBJ to be 3rd person singular, which *us*, *them* etc. are not.[7]

the course of a derivation, but only on final representations, some of which will therefore be invalid. The problem is that the probability mass associated with such representations is lost. Bod and Kaplan (2003) address this issue by re-normalizing to take account of this wasted probability mass.

[7] Another way of thinking about this problem is as an exacerbation of the problem of *data sparsity*: an approach like this will require much more data to get an accurate picture of the contexts where words and phrases can occur. Data sparsity is one of the most pervasive and difficult problems for

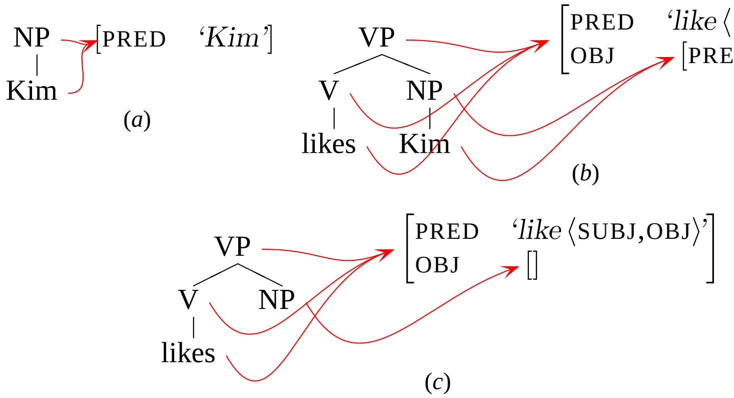


Figure 8: Overgeneral *Discard* fragments

(4) Kim likes Sam.
(5) Sam likes them/us/me/you/the children.

To deal with this problem, B&K introduce a further operation, *Discard*, which produces more general fragments by erasing features. *Discard* can erase any combination of features apart from PRED, and those features whose values ϕ-correspond to remaining c-structure nodes. As regards the fragments *Sam* and *Kim*, this means everything except the PRED can be removed, as in Figure 8 (*a*). In the case of *likes Kim* in Figure 6 (*c*), this means everything can be removed except for the value of PRED and the OBJ (and its PRED), see Figure 8 (*b*). In the case of *likes NP* in Figure 7, it means everything can be removed except the PRED and the OBJ (however, though the OBJ remains, the features it contains can be deleted), see Figure 8 (*c*).

Clearly, such fragments are *over*-general (under specific). For example, the fragment for *Kim* in Figure 8 (*a*) will be able to appear as subject of a non-third person singular verb, as in (6); the fragments for *likes NP* and *likes Kim* will allow non-third singular subjects (and subjects marked accusative), and the fragment for *likes NP* will also allow a nominative object, as in (7).

(6) *Kim were happy.
(7) *Them likes we.

To deal with this, B&K propose a redefinition of grammaticality: rather than regarding as grammatical anything which can be given an analysis, they regard an utterance as grammatical if it can be derived without using *Discard* fragments. For words with relatively high frequency (including common names such as *Kim* and *Sam* and verbs such as *likes*) this is likely to work. For example, every derivation of examples like (6) and (7) is likely to involve *Discard* fragments, so they will be correctly classified as ungrammatical. Equally, (4) will have a non-*Discard*

statistical approaches to natural language.

derivation, and be correctly classified as grammatical, so long as *Kim* appears at least once as a subject, and *Sam* appears at least once as an object, and (5) will have a non-$Discard$ derivation so long as *likes* appears with a sufficiently wide range of object NPs.

The reason this can be expected to work for high frequency words is that for such words the corpus distribution represents the true distribution (i.e. in the language as whole). Unfortunately, most words are *not* high frequency, and their appearance in corpora is not representative of their true distribution. In fact, it is quite common for more than 30% of the words in a corpus to appear only once — and of course this single occurrence is unlikely to reflect the true potential of the word.[8]

For example, in the British National Corpus (BNC) the noun *debauches* ('moral excesses') appears just once, as in (8), where it will be *acc.* Thus, the only way to produce (9) will be to use a $Discard$ fragment. But (8) and (9) are equally grammatical.

(8) [H]e . . . shook Paris by his wild debauches on convalescent leave.
(9) His wild debauches shook Paris.

Similarly, the verbs *to debauch* ('to corrupt morally') and *to hector* ('talk in a bullying manner') appear several times, but never with a first person singular subject: So analyzing (10) and (11) will require $Discard$ fragments, and they will be classified as ungrammatical. But both are impeccable.

(10) I never debauch anyone.
(11) I never hector anyone.

In short: there is a serious theoretical problem with the way LFG-DOP fragments are defined. Without $Discard$, the fragments are *under*general, and the model undergenerates, e.g. it cannot produce (4) and (5). There is a clear need for a method of producing more general fragments via some operation like $Discard$. However, as formulated by B&K, $Discard$ produces fragments that are *over*general, and the model overgenerates, producing examples like (6) and (7). Since B&K's attempt to avoid this problem via a redefinition of grammaticality does not help, we need to consider alternative approaches. The most obvious being to impose constraints on the way $Discard$ operates (cf Way, 1999).[9]

[8] Baroni (to appear) notes that about 46% of all words (types) in the written part of the British National Corpus (90 million tokens) occur only once (in the spoken part the figure is 35%, lower, but still above $1/3$). Of course, the BNC is not huge by human standards: listening to speech at normal rates (say, 200 words per minute) for twelve hours per day, one will encounter more than half this number of tokens each year ($200 \times 60 \times 12 \times 365 = 52,560,000$). But Baroni also observes that the proportion of words that appear only once seems to be largely independent of corpus size.

[9] A number of participants at LFG07 suggested alternative approaches based on 'smoothing', rather than $Discard$ (see also Hearne and Sima'an (2004)). Suppose, we have seen the proper name *Alina* just once, marked *nom* (*Alina*$_{nom}$). We 'smooth' the corpus data, by treating *Alina*$_{acc}$ as an 'unseen event' (e.g. we might assign it a count of 0.5). We can generalize this to eliminate

4 Constraining *Discard*

The problem with B&K's formulation of $Discard$— the reason it produces overgeneral fragments — is that it is indiscriminate. In particular, it does not distinguish between features which are 'inherent' to a fragment (that is, 'grammatically necessary' given its c-structure), and those which are 'contextual' or 'contingent' given its c-structure and are simply artifacts of structure that has been eliminated by the decomposition operations. The former must not be discarded if we are to avoid overgeneration; the latter can, and in the interest of generality should, be discarded. Consider, for example, the fragment for *likes NP* in Figure 7. Intuitively, the PER and NUM features on the object NP are just 'contextual' here — they simply reflect the presence of a third person singular NP in the original representation. On the other hand, the CASE feature on the object is grammatically necessary, as are the PER, NUM and CASE features on the subject NP (given that the verb is *likes*). Similarly, with fragments for NPs like *Sam* and *Kim*: PER and NUM features seem to be grammatically necessary, but CASE seems to be an artefact of the context in which the fragments occur (while with a fragment for *she* all three features would be grammatically necessary).

One approach would be to look for general constraints on $Discard$, e.g. to try to identify certain features as grammatically 'essential' in some way, and immune to $Discard$ (i.e. like PRED for B&K). While appealing, this seems to us unlikely to be sucessful, and certainly no plausible candidates have been proposed.[10]

We think this is not an accident. Rather, the difficulty of finding general constraints on $Discard$ is a reflection of a fundamental feature of f-structures, and LFG: the fact that f-structures do not record the 'structural source' of pieces of f-structure. This is in turn a reflection of an important fact about natural language — one for which constraint based formalisms provide a natural expression: that information at one place in a representation may have many different structural sources (in the case of agreement phenomena, many sources simultaneously). Consider, for

the need for $Discard$: we simply hypothesize similar unseen events for all possible attribute-value combinations. This is an interesting approach, but (a) it will overgenerate, and (b) we will still be unable to reconstruct any idea of grammaticality. To see this, consider that we will also treat *Alina* marked plural (*Alina*$_{pl}$) as an unseen event, and presumably assign it the same count as *Alina*$_{acc}$. We will now be able to derive **Aline run* (so we have overgeneration). Moreover, the same arguments that we used to show the inadequacy of $Discard$ as a basis for a notion of grammaticality apply here, equally (e.g. if we try to identify ungrammaticality with 'involving a smoothed fragment'). Notice it is not the case that grammatical sentences will receive higher probability on such an account: suppose that the probability of *NP run* is the same or higher than *We saw NP*: it is likely that the probability assigned to **Alina run* will be the same or higher than *We saw Alina*. (We are especially grateful to Ron Kaplan, Jonas Kuhn, and Grzegorz Chrupała for stimulating discussion on this point.)

[10] Way (1999), suggests it might be possible to classify features as 'lexical' or 'structural' in some general fashion (so the presence of 'lexical' features in fragments would be tied to the presence of lexical material in c-structures in the same way as PRED). He suggests PER and NUM might be lexical, and CASE might be structural, but notice that there are cases where CASE is associated with particular lexical items (e.g. pronouns *she*, *her*), and where PER and NUM values are associated with a particular structure (e.g. subject of a verb with a third person singular reflexive object, such as *NP criticized herself*).

example, the NUM:*pl* feature that will appear on the subject NPs in the following:

(12) These sheep used to be healthy.
(13) Sam's sheep are sick.
(14) Sam's sheep used to look after themselves.
(15) These sheep are able to look after themselves.
(16) Sheep can live in strange places.

In (12), this feature is a reflex of the plural determiner; in (13) it is a result of the form of the verb (*are*); in (14) it is a result of the reflexive pronoun; in (15) it comes from all these places at once; in (16) it is the *absence* of an article that signals that the noun is plural.

Thus, instead of trying to find general constraints, we propose that the production of generalized fragments should be constrained by the existence of what we will call 'abstract fragments'. Intuitively, abstract fragments will encode information about what is grammatically essential, and so provide an upper bound on the generality of fragments that can be produced by $Discard$. We will call this generalizing operation $cDiscard$ ('constrained $Discard$'). Furthermore, we propose that the knowledge underlying such abstract fragments be expressed using normal LFG grammar rules.

Formally, the key insight is that it is possible to think of a grammar and lexicon as generating a collection of (often very general) fragments, by constructing the minimal c-structure that each rule or lexical entry defines, and creating ϕ-links to pieces of f-structure which are minimal models of the constraints on the right-hand-side of the rule. We will call fragments produced in this way 'basic abstract fragments'.

For example, suppose that, in response to the problems discussed above, we postulate the rules and entries in (17). These rules can be interpreted so as to generate the basic abstract fragments in Figure 9.[11]

(17) a. $\mathrm{S} \rightarrow \begin{array}{cc} \mathrm{NP} & \mathrm{VP} \\ (\uparrow\textsc{subj case})=\textit{nom} & \uparrow=\downarrow \end{array}$

b. $\mathrm{VP} \rightarrow \begin{array}{cc} \mathrm{V} & \mathrm{NP} \\ \uparrow=\downarrow & (\uparrow\textsc{obj case})=\textit{acc} \end{array}$

c. *Kim* NP (↑NUM)=*sg*
(↑PER)=*3*

d. *she* NP (↑NUM)=*sg*
(↑PER)=*3*
(↑CASE)=*nom*

e. *her* NP (↑NUM)=*sg*
(↑PER)=*3*
(↑CASE)=*acc*

[11] Notice that we do not follow the normal LFG convention whereby the absence of f-structure annotation on category is interpreted as '↑=↓': absence of annotation means exactly an absence of f-structure constraints. Notice also that this means we are treating the ϕ-correspondence as a partial function in abstract fragments: in Figure 9 (a) the NP is not linked to any f-structure.

f. *likes* V (↑SUBJ NUM)=*sg*
(↑SUBJ PER)=*3*
(↑TENSE)=*pres*

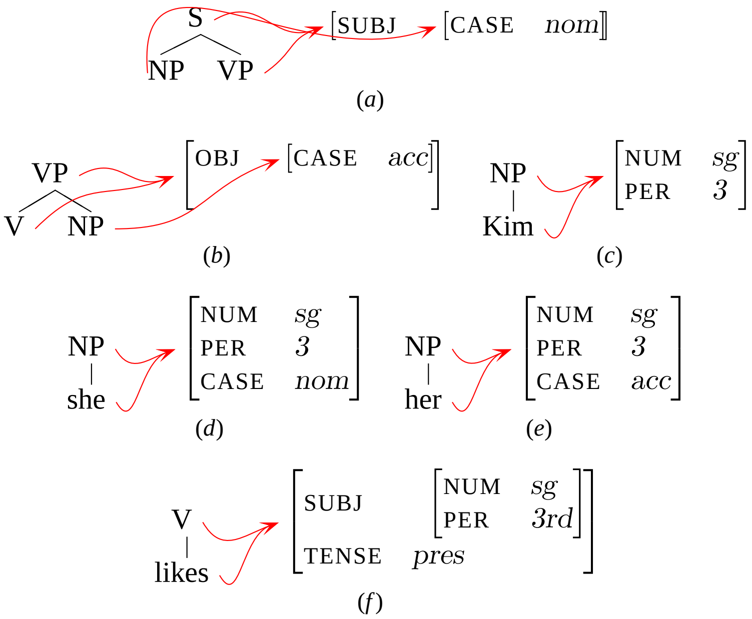


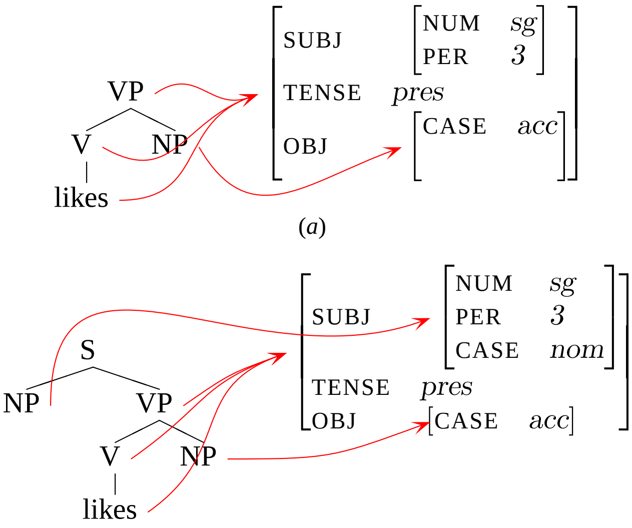
Figure 9: Basic abstract fragments generated by the grammar rules in (17)

Formally speaking, these are fragments in the normal sense, and they can be composed in the normal way. For example composing Figure 9 (*b*) and Figure 9 (*f*) will produce the ‘derived’ abstract fragment in Figure 10 (*a*). This in turn can be composed with Figure 9 (*a*) to produce Figure 10 (*b*). The idea is that such fragments can be used to put an upper bound on the generality of the fragments produced by $cDiscard$, by requiring the latter to be ‘licensed’ by an abstract fragment.

More precisely, we require that, for a fragment f, if $cDiscard(f)$ produces fragment f_d, then there must be some abstract fragment f_a which *licenses* f_d, which for the moment we take to mean f_a ‘frag-subsumes’ f_d. We will say that an abstract fragment f_a *frag-subsumes* a fragment f_d just in case:

1. the c-structures are isomorphic, with identical labels on corresponding nodes; and
2. the ϕ-correspondence of f_a is a subset of the ϕ-correspondence of f_d (recall that ϕ-correspondences are functions, i.e. sets of pairs); and
3. every f-structure in f_a subsumes (in the normal sense) the corresponding f-structure of f_d.[12]

[12] This desciption glosses over a small formal point: normal fragments contain an f-structure with a single root. For abstract fragments this will not always be the case. For example, a rule like



(*b*)

Figure 10: Derived abstract fragments

To see the effect of this, consider the *Root* and *Frontier* fragments in Figure 11 (*b*), (*d*) and (*f*), and the abstract fragments that would license possible applications of *Discard* to them, in Figure 11 (*a*), (*c*) and (*e*).

The abstract fragment in Figure 11 (*a*) will license the discarding of PER and NUM from the object slot of Figure 11 (*b*), but will not permit discarding of TENSE information, or information about the CASE of the subject or object, or PER and NUM information from the subject. Thus, we will have fragments of sufficient generality to analyze (18), but not (19):

(18) Sam likes them/us/me/the children. [=(5)]
(19) *Them likes we. [= (7)]

Similarly, the abstract fragment in Figure 11 (*c*) will license generalized fragments for *Kim* from which CASE has been discarded, but will not allow fragments which from which PER or NUM information has been discarded. Thus, as we would like, we will be able to analyze examples where *Kim* is an object, but not where it is, say, the subject of a non-third person singular verb:

(20) Kim likes Sam. [= (4)]
(21) *Kim were happy. [= (6)]

On the other hand, the abstract fragment in Figure 11 (*e*) will not permit any features to be discarded from *her*, which will therefore be restricted to contexts which allow third person singular accusatives:

S →NP VP (without any constraints) should produce an abstract fragment with c-structure consisting of three nodes, each associated with a separate, empty, f-structure.

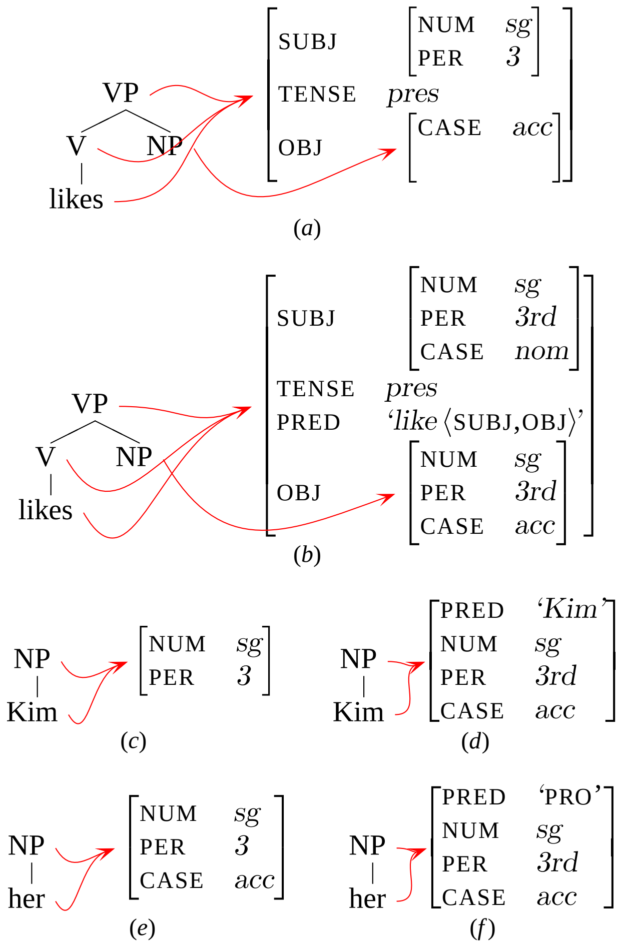


Figure 11: *Root*, *Frontier*, and abstract fragments

(22) Sam likes her.

(23) *Her likes Sam.

5 General Constraints

The previous section has shown how one source of overgeneration can be avoided. A second source of overgeneration arises from the fact that, while it provides a reasonable model of normal c- and f-structure constraints (i.e. defining equations), an LFG treebank is only a poor reflection of other kinds of constraint, e.g. negative constraints, functional uncertainty constraints, existential constraints, and constraining equations.[13] A treebank is a finite repository of positive information, and cannot properly reflect negative constraints, constraints with potentially infinite

[13] See Dalrymple (2001) for discussion and exemplification of such constraints.

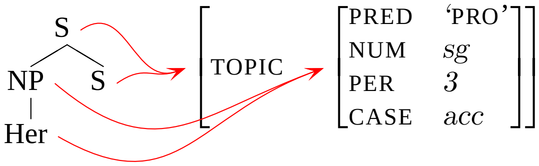


Figure 12: A *cDiscard Frontier* fragment

scope, or constraints whose essential purpose is information ‘checking’. In this section we will show how the approach of the previous section can be extended to address this source of overgeneration. For reasons of space, we will focus on functional uncertainty constraints and negative constraints.

As an example of a functional uncertainty constraint, consider the need to ‘link’ topicalized constituents. Suppose the treebank contains representations of examples like (24) and (25).

(24) Her, Sam likes.
(25) Her, we think Sam likes.

As things stand, it will be possible to produce a fragment like Figure 12 from (24) by deleting the structure corresponding to *Sam likes* (and discarding a number of features like TENSE, which are not relevant here). Notice it will be possible to compose any complete sentence with this, and so derive ungrammatical examples like the following, in which the topicalized constituent *her* is not linked to any normal grammatical function.

(26) *Her, Sam likes Kim.

In a normal LFG grammar, examples like (26) are excluded by including a functional uncertainty constraint on the rule that produces topicalized structures:[14]

(27) $$\begin{array}{lcc} \text{S} \rightarrow & \text{NP} & \text{S} \\ & (\uparrow\text{TOPIC})=\downarrow & \uparrow=\downarrow \\ & (\uparrow\text{COMP* GF})=\downarrow & \end{array}$$

As things stand, the LFG-DOP model is unable to prevent examples like (26) being derived: there is no way of capturing the effect of anything like an uncertainty constraint.

As regards negative constraints, in Section 4 we expressed facts about subject verb agreement with *likes* by means of a positive constraint requiring its subject to

[14] In (27), GF is a variable over grammatical function names, such as OBJ and SUBJ, and COMP* is a regular expression meaning any number of COMPs (including zero). COMP is the grammatical function associated with complement clauses. Thus, the constraint requires the NP’s f-structure to be the OBJ (or SUBJ, etc.) of its sister S, or of a complement clause inside that S, or a complement clause inside a complement clause (etc).

be 3rd person singular. This still leaves the problem of agreement for other forms. For example, we must exclude *like* appearing with a 3rd person singular form, as in (28).

(28) *Sam like Kim.

This can be expressed with a disjunction of normal constraints, but the most natural thing to say involves a negative constraint, along the lines of (29) (which simply says that the subject of *like* must not be third person singular). The existing apparatus provides no way of encoding anything like this.

(29) *like* V $\neg\Big($ ($\uparrow$SUBJ PER)=*3* ($\uparrow$SUBJ NUM)=*sg* $\Big)$

In fact, apparatus to avoid this sort of overgeneration is a straightforward extension of the approach described above.

- We add to fragments a fourth component, so they become 4-tuples: $\langle c, \phi, f, \mathit{Constr}\rangle$, where *Constr* is a collection of ‘other’ (i.e. non-defining) constraints.
- For basic abstract fragments the elements of Constr are the ‘other’ constraints required by the corresponding rule or lexical entry.
- Combining abstract fragments involves unioning these sets of constraints.
- Licensing a fragment involves adding these constraints to the fragment (i.e. fragments inherit the Constraints of the abstract fragment that licenses them).
- The composition process is amended so as to include a check that these constraints are not violated (specifically, we require that, in addition to normal completeness and coherence requirements, the f-structure of any final representation we produce must satisfy all constraints in Constr).

The idea is that, given a grammar rule like (29), any basic abstract fragment for *like* will include a negative constraint on the appropriate f-structure, which will be inherited by any derived abstract fragment, and any fragment that is thereby licensed. So, for example, the most general $\mathit{cDiscard}$ fragment for *NP like Kim* will be as in Figure 13. While it will be possible to adjoin a 3rd person singular NP to the subject position of this fragment, this will not lead to a valid final representation, because the negative constraint will not be satisfied. Thus, as one would hope, we will be able to derive (30), but not (31).

(30) They like Kim.
(31) *Sam like Kim.

Similarly, the rule in (27) will produce abstract fragments which contain the uncertainty constraint given, and these will license normal fragments like that in Figure 14. Again, the only valid representations which can be constructed which satisfy this constraint will be ones which contain a ‘gap’ corresponding to the TOPIC. That is, as one would like, we will be able to produce (32), but not (33):

(32) Her, Sam (says she) likes.
(33) *Her, Sam (says she) likes Kim.

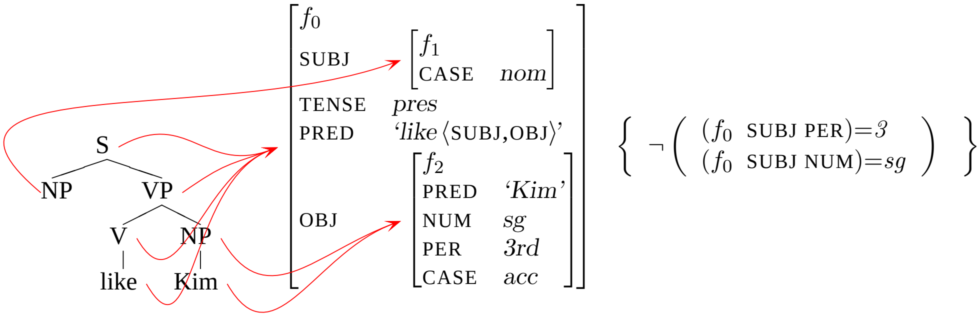


Figure 13: Fragment incorporating a negative constraint

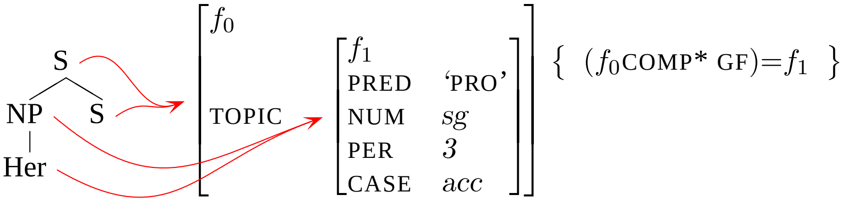


Figure 14: Fragment incorporating an uncertainty constraint

6 Discussion

The proposals presented in the previous sections constitute a relatively straightforward extension to the formal apparatus of LFG-DOP, but they are open to a number of objections, and they have theoretical implications of wider significance.

One kind of objection that might arise is a result of the relatively minor phenomena we have used for exemplification (case assignment and person-number agreement in English). This objection is entirely misplaced. First, because, in an LFG context, similar problems will arise in relation to any phenomenon whose analysis involves f-structure attributes and values. More generally, similar problems of fragment generality will arise whenever one tries to generalize DOP approaches beyond the context-free case, e.g. to deal with semantics.[15] More generally still, analogues of the problems we have identified with fragment generality and capturing the effect of 'general' constraints on the basis of a finite collection of example representations will arise with any 'exemplar' based approach.

A second source of objections might arise from the fact that we have focused

[15] At least, this is the case if one wants to preserve the idea that a treebank consists of representations in the normal sense. In the approach to semantic interpretation in DOP described in Bonnema et al. (1997) these problems are avoided at the cost of not using semantic representations in the normal sense. Rather than having semantic representations, the nodes of trees are annotated with an indication of how the semantic formula of the node is built up from the semantic formulae of its daughters, and hence how it should be decomposed. The 'fragment generality' problem is sidestepped by explicitly indicating on each and every node how its semantic representation should be decomposed as fragments are created.

on the problem of overgeneration: one might object (a) that in a practical, e.g. language engineering, setting this is not very important, and (b) that in a probabilistic setting, such as DOP, overgeneration can be hidden statistically (e.g. because ungrammatical examples get much smaller probability compared to grammatical ones).

As regards (a), the appropriate response is that a model which overgenerates is generally one which assigns excessive ambiguity (which is a pervasive problem in practical settings). Sag (1991) gives a large number of plausible examples. In relation to subject-verb agreement, he notes that the following are *un*ambiguous, but will be treated as ambiguous by any system that ignores subject-verb agreement: (34) presumes the existence of a unique English-speaking Frenchman among the programmers; (35) presumes there is a unique Frenchman among the English speaking programmers.

(34) List the only Frenchman among the programmers who understands English.
(35) List the only Frenchman among the programmers who understand English.

Similarly, a system which does not insist on correct linking of Topics will treat (36) and (37) as ambiguous, when both are actually unambiguous (in (36) *to them* must be associated with *contributed*, in (37) it must be associated with *appears*, because *contribute* requires, and *discover* forbids, a complement with *to*):

(36) To them, Sam appears to have contributed it.
(37) To them, Sam appears to have discovered it.

As regards (b), it is important to stress that the problem of overgeneration as we describe it has to do with the characterization of grammaticality (i.e. the characterization of a language), and grammaticality simply cannot be identified with relative probability (casual inspection of almost any corpus will reveal many simple mistakes, which are uncontroversially ungrammatical, but have much higher probability than perfectly grammatical examples containing, e.g., rare words).

A third objection would be that in avoiding overgeneration, we have also lost the ability to deal with ill-formed input (robustness). But there is no reason why the model should not incorporate, in addition to 'constrained $Discard$', an unconstrained operation like the original B&K $Discard$. Notice that this would now give a correct characterization of grammaticality (a sentence would be grammatical if and only if it can be derived without the use of unconstrained $Discard$ fragments).

A fourth, and from a DOP perspective very natural, objection would be that these proposals in some sense violate the 'spirit' of DOP — where an important idea is exactly to dispense with a grammar in favor of (just) a collection of fragments. A partial response to this is to note that to a considerable degree the sort of grammar we have described is implicit in the original treebank. For example, the set of c-structure rules can be recovered from the treebank by simply extracting all trees of depth one. This will produce a grammar without f-structure constraints, and abstract fragments with empty f-structures and constraint sets, which is exactly equivalent to the original B&K model. Taken as a practical proposal for grammar engineering, the idea would be that one can begin with such an unconstrained model, and simply add constraints to these c-structure rules to rule out overgeneration. This can clearly be done incrementally, and in principle, the full range of LFG rule notation should be available, so this should be a relatively straightforward and natural task for a linguist. It should be, in particular, much easier than writing a normal grammar.

However, it is also possible to take the proposal in a different way, 'theoretically', as describing an idea about linguistic knowledge, and human language processing and acquisition. Taken in this way, the suggestion is that a speaker has at her disposal two knowledge sources: a database of fragments (in the normal DOP sense), which one might think of as a model of grammatical usage, and a grammar (an abstract fragment grammar) which expresses generalizations over these fragments, which one might take to be a characterization of something like grammatical competence. Notice that on this view: (i) the grammar as such plays no role in sentence processing (but only in fragment creation, i.e. off-line); (ii) the task of the learner is only secondarily to construct a grammar (the primary task is the creation of the fragment database — learning generalizations over this is a secondary task); (iii) the grammar does not generate or otherwise precisely characterize the language (this is achieved by the fragment database with the composition operation), rather its job is to license or legitimize the fragments in the fragment database. Taken in this way, the model is an enrichment of the standard DOP approach.

References

Arnold, Doug and Linardaki, Evita. 2007. A Data-Oriented Parsing Model for HPSG. In Anders Søgaard and Petter Haugereid (eds.), *2nd International Workshop on Typed Feature Structure Grammars (TFSG'07)*, pages 1–9, Tartu, Estonia: Center for Sprogteknologi, Kobenhavens Universitet, Working Papers, Report No. 8.

Baroni, Marco. to appear. Distributions in Text. In Anke Lüdeling and Merja Kytö (eds.), *Corpus Linguistics: An International Handbook*, Berlin: Mouton de Gruyter.

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

Bod, Rens. 2000a. An Empirical Evaluation of LFG-DOP. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING00)*, volume 1, pages 62–68, Saarbrüken.

Bod, Rens. 2000b. An Improved Parser for Data-Oriented Lexical-Functional

Analysis. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 61–68, Hong Kong.

Bod, Rens. 2006. Exemplar-Based Syntax: How to Get Productivity From Examples. *The Linguistic Review* 23(3), 291–320, (Special Issue on Exemplar-Based Models in Linguistics).

Bod, Rens and Kaplan, Ronald. 1998. A Probabilistic Corpus-Driven Model for Lexical Functional Analysis. In *Proceedings of COLING-ACL '98: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume 1, pages 145–151, Montreal, Canada.

Bod, Rens and Kaplan, Ronald. 2003. A DOP Model for Lexical-Functional Grammar. In Rens Bod, Remko Scha and Khalil Sima'an (eds.), *Data-Oriented Parsing*, Chapter 12, pages 211–233, Stanford, California: CSLI Publications.

Bod, Rens and Scha, Remko. 1997. Data Oriented Language Processing. In Steve Young and Gerrit Bloothooft (eds.), *Corpus-Based Methods in Language and Speech Processing*, volume 2 of *Text, Speech and Language Technology*, pages 137–173, Dordrecht: Kluwer Academic Publishers.

Bod, Rens, Scha, Remko and Sima'an, Khalil (eds.). 2003. *Data-Oriented Parsing*. Stanford, California: CSLI Publications.

Bonnema, Remko, Bod, Rens and Scha, Remko. 1997. A DOP Model for Semantic Interpretation. In *Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the EACL*, pages 159–167, Madrid, Spain.

Bonnema, Remko, Buying, Paul and Scha, Remko. 1999. A new probability model for Data Oriented Parsing. In Paul Dekker and Gwen Kerdiles (eds.), *Proceedings of the 12th Amsterdam Colloquium*, pages 85–90, Amsterdam, The Netherlands.

Bresnan, Joan (ed.). 1982. *The Mental Representation of Grammatical Relations*. Cambridge, Massachussets: MIT Press.

Bresnan, Joan. 2001. *Lexical-Functional-Syntax*. Blackwell Textbooks in Linguistics, Oxford: Blackwell.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*. New York: Academic Press.

Dalrymple, Mary, Kaplan, Ronald M., Maxwell, John T. and Zaenen, Annie (eds.). 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, California: CSLI Publications.

Finn, Riona, Hearne, Mary, Way, Andy and van Genabith, Josef. 2006. GF-DOP: Grammatical Feature Data-Oriented Parsing. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG06 Conference*, Stanford, California: CSLI Publications.

Hearne, Mary and Sima'an, Khalil. 2004. Structured Parameter Estimation for LFG-DOP. In N. Nicolov, K. Bontcheva, G. Angelova and R. Mitkov (eds.), *Recent Advances in Natural Language Processing III: Selected papers from the RANLP 2003*, volume 260 of *Current Issues in Linguistic Theory*, pages 183–192, Amsterdam: John Benjamins.

Johnson, Mark. 2002. The DOP Estimation Method is Biased and Inconsistent. *Computational Linguistics* 28, 71–76.

Jurafsky, Dan. 2003. Probabilistic Modeling in psycholinguistic comprehension and production. In Rens Bod, Jennifer Hay and Stefanie Jannedy (eds.), *Probabilistic Linguistics*, Chapter 3, pages 39–96, Cambridge, Massachusetts: MIT Press.

Linardaki, Evita. 2006. *Linguistic and statistical extensions of Data Oriented Parsing*. PhD thesis, University of Essex.

Neumann, Günter. 2003. A Data-Driven Approach to Head-Driven Phrase Structure Grammar. In Rens Bod, Remko Scha and Khalil Sima'an (eds.), *Data-Oriented Parsing*, Chapter 13, pages 233–251, Stanford, California: CSLI Publications.

Sag, Ivan A. 1991. Linguistic Theory in Natural Language Processing Language Processing. In Ewan Klein and Frank Veltman (eds.), *Natural Language and Speech*, pages 69–84, Berlin: Springer Verlag.

Sima'an, Khalil and Buratto, Luciano. 2003. Backoff Parameter Estimation for the DOP Model. In Nada Lavrac, Dragan Gamberger, Hendrik Blockeel and Ljupco Todorovski (eds.), *Proceedings of the European Conference on Machine Learning*, Lecture Notes in Artificial Intelligence, pages 373–384, Berlin: Springer.

Way, Andy. 1999. A Hybrid Architecture for Robust MT using LFG-DOP. *Journal of Experimental and Theoretical Artificial Intelligence* 11, 441–471, (Special Issue on Memory-Based Learning).

Zollmann, Andreas. 2004. *A Consistent and Efficient Estimator for the Data-Oriented Parsing Model*. Masters Thesis, ILLC, Amsterdam, The Netherlands.