

# A Data-Oriented Parsing Model for HPSG

**Doug Arnold** and **Evita Linardaki**

Language and Linguistics

University of Essex

Wivehoe Park, Colchester, UK

{doug, elinaro}@essex.ac.uk

## Abstract

Data Oriented Parsing (DOP) is based on the idea of processing new input by combining fragments (associated with some probabilities) that are extracted from a treebank. In the simplest case these fragments are subparts of simple phrase structure trees (Tree-DOP). The approach is attractive in many ways but the impoverished representational basis is a serious drawback from a linguistic point of view. This paper describes the theoretical foundations of a novel version of DOP with a richer representational basis, a form of Typed Feature Structure Grammar, specifically Head-driven Phrase Structure Grammar (HPSG).

## 1 Introduction

The development of probabilistic extensions of normal grammatical theories has both practical and theoretical motivations. On the practical side, a way of incorporating probabilistic techniques seems indispensable for efficient implementations and applications. On the theoretical side, there is large, and growing, evidence of probabilistic effects in human language processing.

One approach to this involves associating the rules of a competence grammar with probabilities computed from syntactically annotated corpora. But simply adding probabilities to rules is of limited

value, because it cannot express probabilistic dependencies that extend outside the locality of the individual rule. Data Oriented Parsing (DOP) models (e.g. Bod (2003); Bod (1995); Bod et al. (2003)) overcome this limitation, allowing probabilities to be associated with arbitrarily large syntactic constructions — tree fragments extracted from a treebank. In the simplest case (‘Tree-DOP’) these fragments are subparts of context free phrase structure trees produced by two decomposition operations: *Root* and *Frontier*. *Root* creates ‘passive’ (closed, complete) fragments by extracting substructures, as in Figure 1(b), while *Frontier* produces ‘active’ (open, incomplete) fragments by deleting pieces of substructure, as in Figure 1(c). Probabilities are assigned to such fragments based on their relative frequency. Conceived abstractly, parsing in DOP involves a process of *derivation*: a DOP derivation is a process of combining fragments (with their associated probabilities), so that the open slots in active fragments become filled. Disambiguation involves finding the structure(s) with the highest probability.

The approach is attractive in many ways (not least, it is a very *elegant* framework for probabilistic grammar), but the impoverished representational basis is a serious drawback from a linguistic point of view. Bod and Kaplan (1998) address this issue by proposing a linguistically richer version of DOP which uses Lexical Functional Grammar (LFG) representations. Though LFG-DOP constitutes a very powerful model of language performance, it also suffers from several disadvantages. The first of

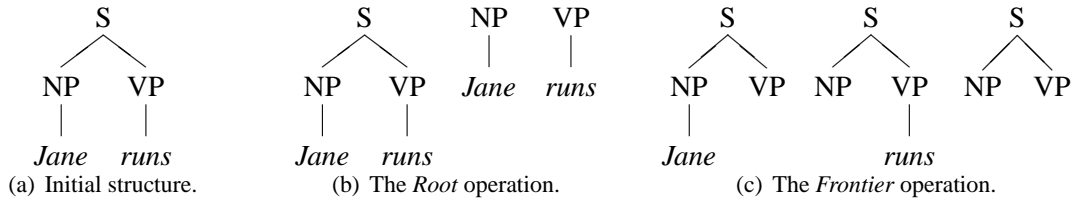


Figure 1: Decomposition of *Jane runs* in Tree-DOP.

these relates to the *generality* of the fragments it involves. The LFG-DOP versions of the standard decomposition operations create fragments that are over-specific, leading to under-generation (and exacerbating the normal problem of data sparsity). For example, from the corpus representation of *Jane runs*, LFG-DOP *Root* and *Frontier* will produce fragments containing category and feature information along the lines of (1) and (2).

- (1)  $[_{NP} \text{Jane}]_{3rd/sg/fem/nom}$
- (2)  $[_{S} \text{NP}_{3rd/sg/fem/nom} \text{runs}]$

These fragments will not be usable in parsing either (3), which involves *Jane* appearing as an object, hence marked accusative, whereas the fragment in (1) is nominative; or (4), because *runs* in (2) requires its subject to be feminine, and *Jack* is presumably masculine.

- (3) Sam likes Jane.
- (4) Jack runs.

To overcome this, Bod and Kaplan formulate a third decomposition operation known as *Discard*, which generalizes over the fragments produced by the other two. *Discard*, however, applies in a highly unconstrained manner causing (i) the size of the fragment database to explode and (ii) allowing *under-specific* fragments such as (5) and (6) to be produced, leading to overgeneration, as with examples like (7) and (8).

- (5)  $[_{NP} \text{Jane}]_{fem/nom}$
- (6)  $[_{S} \text{NP}_{3rd/sg} \text{runs}]$

- (7) \*Jane run.
- (8) \*Him runs.

To deal with this, Bod and Kaplan propose a revised definition of grammaticality: ‘grammatical’ sentences are those that can be derived without use

of *Discard* fragments. But this is unsatisfactory from a theoretical point of view, because it may easily classify as ungrammatical impeccable examples like (3) and (4).<sup>1</sup>

Of course, what we would like is to produce fragments that are more general than (1) and (2), but not over-general like (5) and (6). Fragments more like (9) and (10) are preferable, where only ‘inherent’, grammatically necessary, features remain, and features that are purely the result of the context in which the fragment appeared (nominative marking on *Jane*, and feminine on the subject slot of *runs*) have been removed. But the LFG-DOP decomposition operations do not deliver this.

- (9)  $[_{NP} \text{Jane}]_{3rd/sg/fem}$
- (10)  $[_{S} \text{NP}_{3rd/sg/nom} \text{runs}]$

A further problem is that not all of LFG’s well-formedness conditions can be checked during the derivation process. In particular, LFG uses non-monotonic operations which have to be applied as filters on final representations. The problem here is that some probability mass will be assigned to structures that are filtered out, and this probability mass is wasted, it ‘leaks away’, with the result that the values assigned to structures are not genuine probabilities.<sup>2</sup>

In the following section we set out the theoretical foundations of a novel version of DOP based on a form of Typed Feature Structure Grammar, specifically Head-driven Phrase Structure Grammar

<sup>1</sup>Suppose, for example, that *runs* only ever occurs with a feminine subject in the corpus, and *Jane* is never an object, hence never accusative. For such common words, this is perhaps implausible, but it is very likely for less common words and phrases.

<sup>2</sup>This problem is not fatal, since the values can be ‘renormalized’ to compensate for lost probability mass (Bod and Kaplan, 2003), but this solution is not very aesthetically appealing.

(HPSG) (Pollard and Sag, 1994; Ginzburg and Sag, 2000), that addresses these issues.<sup>3</sup>

## 2 HPSG-DOP

Presenting a DOP model involves instantiating the following four parameters: (i) how utterances are represented; (ii) how representations are decomposed into fragments; (iii) how fragments are combined; and (iv) how choices among alternative analyses can be made (disambiguation).

### 2.1 Representation

The representational framework we assume for HPSG-DOP is conventional HPSG, along the lines of Ginzburg and Sag (2000): words and utterances are modeled by feature structures representing *signs*. We will draw feature structures either as Directed Acyclic Graphs (DAGs) or as Attribute Value Matrices (AVMs), using a wide range of standard abbreviations (e.g. ‘NP’ stands for a nominal phrase with empty *SPR* and *COMPS* lists, ‘NP<sub>nom.3</sub>’ is an NP whose *CASE* value is *nom*, and whose *INDEX|PER* value is 3). Figure 2(a) gives the DAG representation of *Jane runs*, while Figure 2(b) presents it as an AVM (both somewhat simplified).

We assume representations are *totally well-typed* feature structures. A representation is *totally well-typed* if all and only the required attributes are present and each of them has an appropriate value. Of course, this only makes sense against the background of a particular *type theory*, that is, a *signature*, which defines an hierarchy of *types*, and a collection of *type constraints* which indicate what combinations of attributes and values are permitted for

<sup>3</sup>This is not the first attempt to define such a model. In particular, Neumann (1999), Neumann (2003) describes an approach which involves extracting a Stochastic Lexicalized Tree Grammar (SLTG) from a parsed HPSG corpus and using it in a manner similar to Tree-DOP. Node are labeled to reflect the HPSG rule-schema that licenses the subtree. A complete parse tree can be unfolded into an HPSG representation by expanding the rule labels and lexical types to the corresponding feature structures. Despite the very different linguistic theories involved, the approach is in many ways very similar to LFG-DOP, and suffers from the same problems. Moreover, the use of trees as the basic representational framework is not entirely consistent with the formal foundations of HPSG, where tree structures have no role as such.

<i>phrase</i>	
PHON	<i>list-of-phonemes</i>
SYNSEM	<i>synsem</i>
DTRS	<i>list-of-signs</i>
HD-DTR	<i>sign</i>

Figure 3: TypeExp(*phrase*)

different types. Fragments should respect the same principles as the representations they are produced from: i.e. they should be *totally well-typed* feature structures. The *total well-typedness* requirement implies that fragments may be subject to a form of type inference which we will refer to as *type expansion*.

**Definition 2.1** (*TypeExp*). *Let  $F$  be a feature structure, and  $T$  a type theory, then  $TypeExp_T(F)$  is the most general and totally well-typed extension of  $F$  according to  $T$  such that  $F \sqsubseteq TypeExp_T(F)$ .*

To take a simple example, type expanding the sort *phrase* produces the feature structure in Figure 3 (assuming the type theory in Ginzburg and Sag (2000)). Type expansion can also produce far more extensive results. For example, there is a type constraint on *hd-subj-strs* (head-subject-structures) that requires re-entrance between the head daughter’s *SUBJ* value and the *SYNSEM* value of the non-head daughter. Moreover, since *hd-subj-str* is a sub-sort of *hd-str*, such structures must also satisfy a type constraint requiring a re-entrance between its *HEAD* value and that of its *HDDTR*. Similarly, type constraints associated with a third person singular verb form like *runs* will require that its subject is *3.sg*.<sup>4</sup>

### 2.2 Decomposition Operations

Decomposition is carried out by versions of *Root* and *Frontier*. These operations are normally defined on trees, to adapt them for feature structures we introduce the notion of the *descendants* (of a *sign*), defined to be the elements of the *sign*’s *DTRS* list, and their descendants (recursively).

We begin with the definition of *Frontier*, which

<sup>4</sup>It is perhaps worth emphasizing the centrality of the type theory in recent versions of HPSG, where essentially a grammar (including the lexicon) consists of nothing but a type theory (this was not the case with in earlier versions of HPSG such as Pollard and Sag (1994), where a variety of other devices were used — these devices have generally been either discarded or reformulated within the type system).

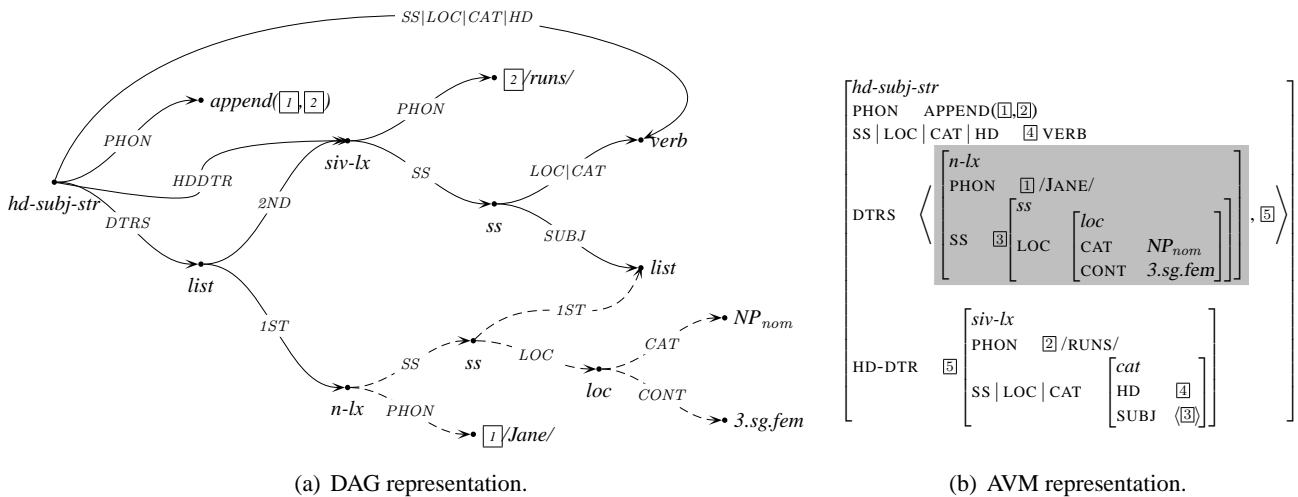


Figure 2: Two representations of *Jane runs*.

produces fragments with open slots, to which other fragments may be attached in the course of derivation.

**Definition 2.2** (*Frontier*). *Frontier erases any combination of a fragment’s descendants and type expands the result, marking the erasure points as ‘open slots’ for composition.*

Consider Figure 2, where one of the descendants (in fact the first daughter) is the *n-lx* (noun lexeme) whose phonology is /Jane/. If *Frontier* applies to this, it will first erase the entire subgraph of Figure 2(a) which is the value of  $DTRS|IST$ , and mark the erasure point with \*, as shown in Figure 4. Notice that this has in effect removed *all* information about the subject of *runs*. This structure must now be type-expanded.

Of course, just how type-expansion works will depend on the precise type-theory involved, but on standard assumptions, an HPSG grammar of English will require that what fills the  $DTRS|IST$  slot is an object of type *sign*, whose *PHON* is re-entrant with the first part of the *PHON* of the whole sentence (i.e. tag 1). This follows from general constraints on phrases, including *head-subj-str*. In addition, the *sign*’s *SYNSEM* (*SS*) value will be re-entrant with the the  $SUBJ|IST$  slot of *runs*. This will restricts it to being a 3rd person, singular, nominative nominal. But there will of course be no constraint that requires the subject of *runs* to be *fem*. That is, type-expansion of Figure 4 can be expected to produce something like

Figure 5.

Notice that this version of *Frontier* produces fragments of the desired level of generality: general enough to allow both masculine and feminine subjects, and so allow *Jack runs*, but not sufficiently general to allow accusative subjects, as in *\*Him runs*.<sup>5</sup>

We now turn to the definition of *Root*, which requires some extra terminology. Let  $F$  be a feature structure with a descendant  $D_F$ . Suppose  $D_F$  is removed from  $F$  giving rise to  $F'$ . Then *Context* ( $D_F$ ) denotes the subgraph rooted at the removal node in  $TypeExp_T(F')$ , and *Inherent* ( $D_F$ ) denotes the relative complement of  $D_F$  and *Context* ( $D_F$ ).

An example will clarify this. Suppose  $F$  is the structure in Figure 2 and  $D_F$  is its *n-lx* *DTR* (i.e. the structure that corresponds to *Jane*, and which we erased with *Frontier* above). As already discussed, if  $D_F$  is removed from  $F$  it gives rise to a structure  $F'$  like the one in Figure 4, and as we have seen  $F'$  will be type-expanded to Figure 5.  $D_F$  is the graph that has been erased in Figure 4, *Context* ( $D_F$ ) is the graph that ‘grows back’ when Figure 4 is type expanded — the part of Figure 5 which is drawn with dotted lines. Intuitively, it is the ‘contextual’ information in the substructure — the information

<sup>5</sup>This formulation of *Frontier* can eliminate any and all descendants of a sign. This is perhaps too liberal. An alternative would be to limit *Frontier* so that it produces fragments containing a lexical head. This possibility is discussed in (Linar-daki, 2006).

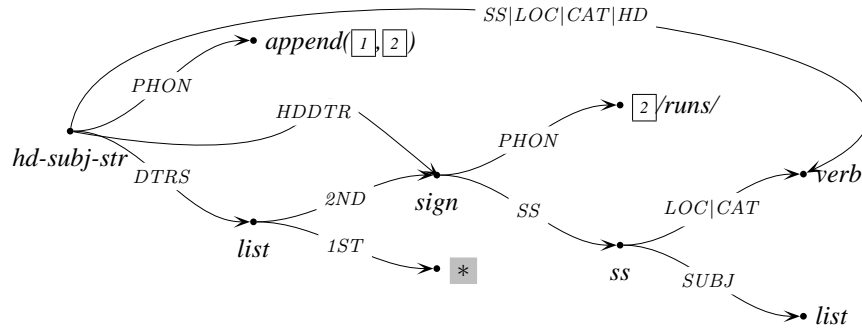


Figure 4:  $F'$ : Erasing  $D_F$ .

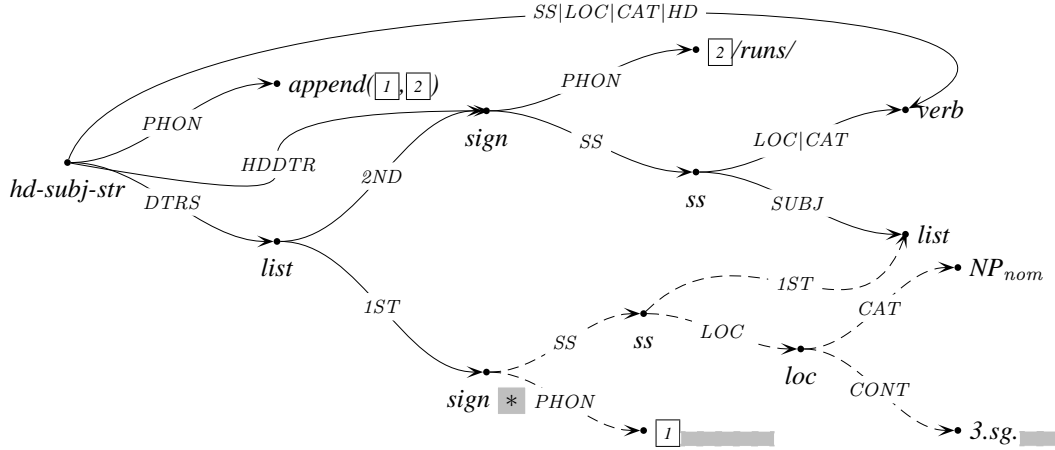


Figure 5:  $TypeExp_T(F')$ .

that could come from the surrounding context. For example, it contains the information that this structure is a 3rd singular nominative NP.

We can get the ‘inherent’, non-contextual, information in this structure by comparing the contextual information with the original. The purely inherent information is just the information that is in the original but not in the contextual information — the relative complement of  $D_F$  and  $Context(D_F)$ . In this example, the purely inherent information includes the fact that is a noun lexeme ( $n-lx$ ), as well as phonological and semantic features of the structure, including the fact that its phonological content is  $/Jane/$  and that it is feminine. This is given as an AVM in Figure 6. Of course, this may exclude some information which should intuitively be regarded as inherent, in the sense of being grammatically necessary (specifically, this will happen if such information is *also* contextual). However, type-expansion can be used to recover this information.

*Root* can now be defined simply:

**Definition 2.3** (*Root*). Given a fragment  $F$ , *Root* selects any descendant  $D_F$  of  $F$  and returns  $TypeExp(Inherent(D_F))$ .

For example, if *Root* applies to *Jane* (i.e. the value of  $DTRS|1ST$  in Figure 2), it will return the type-expansion of the structure in Figure 6. Again, the precise details will depend on the precise type theory (i.e. the grammar and lexicon), but it is plausible that type-expanding  $n-lx$  in a way that is consistent with a  $PHON$  value of  $/Jane/$  will produce something which is 3rd person singular, represented as an AVM in Figure 7. Equally, it is plausible that it will *not* produce a structure that is marked as nominative (there will be nothing in the type system to require

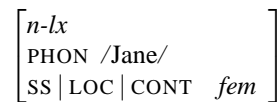


Figure 6: *Inherent* (Jane)

arbitrary nouns to be so marked).<sup>6</sup>

Notice again that this produces a fragment of the desired level of generality: the fragment for *Jane* in Figure 7 is *3.sg.fem*, but not *nom*, as we wanted.

The example we have discussed involves only subject-verb agreement and nominative marking, and may seem rather trivial. It is important to emphasize that because all linguistic phenomena are described using the same apparatus in HPSG (i.e. using the type theory), these decomposition operations can be expected to produce fragments of appropriate generality regardless of the grammatical phenomena involved (subcategorization, raising, extraposition, inversion, long-distance dependencies, etc. etc.). Notice also that because this approach makes no use of a *Discard* operation to generalize fragments, it avoids the explosive effect this operation has on the size of the fragment database that we noted in relation to LFG-DOP.

### 2.3 Head-driven Composition

Composition in HPSG-DOP involves unifying an ‘open’ slot in one with the root of another. For example, composing the fragment in Figure 5 with the fragment corresponding to *Jane* (Figure 7) involves unifying Figure 7 with the node marked \* in Figure 5 (this of course gives back the original representation of *Jane runs*, Figure 2).

Notice that this process is guaranteed to produce only valid structures (i.e. ones which respect the relevant type system). To see this, notice that assuming initial corpus representations are valid, then ‘basic’ fragments produced by decomposing corpus repre-

<sup>6</sup>Applying *Root* to the subject of *She runs* would be similar, except that nominative case *would* be among the features that ‘grows back’ under type-expansion, since it is presumably required by the lexical type that corresponds to *she*.

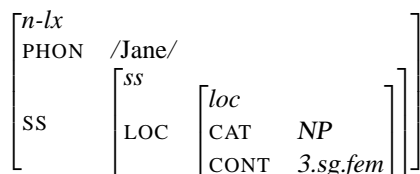


Figure 7:  $TypeExp_T(Inherent(Jane))$ , i.e. *Root* applied to *Jane*.

sentations will also be valid (because they are produced by type-expansion, which by definition preserves validity). But composition involves nothing more than unification, and unifications involving valid structures will only succeed if the result is also valid.

Standard composition approaches in DOP are rightwards or incrementally rightwards directed (Bod, 1995; Neumann, 2003). In an HPSG context, it is interesting to consider a ‘head-driven’ approach to composition, whereby it is the head chain of the initial fragment of a derivation that identifies the order in which open nodes are filled. More specifically, the open nodes of a fragment are filled in such a way that each node along the path leading upwards from the head lexical anchor to the root of the fragment is closed before the next node along the path is considered (so, for example, in the case of a transitive sentence with open subject and object slots, the object slot will be filled first). Composition is thus bidirectional with the direction being identified at each step (rather than in some predefined manner). Such a process is of course reminiscent of head-driven parsing strategies (Proudian and Pollard, 1985; van Noord, 1997, for example).<sup>7</sup>

### 2.4 Fragment Probabilities

As in other DOP models, an HPSG-DOP representation will typically have many different derivations, and any string may have many different representations. Assuming composition steps are treated as independent events, the probabilities of a derivation  $d$  involving the composition of a collection of fragments  $\langle f_1, \dots, f_n \rangle$  can be given as the product of the individual fragment probabilities, as in (11). The probability of a final representation  $R$  with  $m$  derivations  $d_j$  can be given as the sum over the probability of all  $m$  derivations, as in (12).

$$(11) \quad P(d) = \prod_{i=1}^n P(f_i)$$

$$(12) \quad P(R) = \sum_{j=1}^m P(d_j)$$

<sup>7</sup>This is described in more detail in (Linardaki, 2006).

In Tree-DOP, fragment probabilities are calculated using a relative frequency estimator, as in (13): the probability of a fragment  $f_i$  is its frequency of occurrence relative to all fragments  $f$  with the same root node label.

$$(13) \quad P(f_i) = \frac{|f_i|}{\sum_{\text{root}(f)=\text{root}(f_i)} |f|}$$

This is a reasonable definition in the context of Tree-DOP, because the root node label does indeed characterize the combinatory potential of a fragment. However, applying this approach to model with richer representations like HPSG-DOP is problematic.

Suppose, for the sake of argument, we take the *CAT* values of HPSG-DOP fragments as determining their combinatory potential (this includes *HEAD* features and various subcategorization features). The probability of an HPSG-DOP fragment  $f_i$  would then be defined as in (14) — the relative frequency of  $f_i$  to all fragments with the same *CAT* value.

$$(14) \quad P(f_i) = \frac{|f_i|}{\sum_{\substack{v(\text{SS|LOC|CAT},r(f))= \\ v(\text{SS|LOC|CAT},r(f_i))}} |f|}$$

The problem is that this is not guaranteed to identify a probability distribution over the set of valid representations — the combinatory potential of a fragment is not entirely determined by its *CAT* value — there are many other values that may be relevant (e.g. *CONTENT*, and *NONLOCAL* values). This means that a fragment may form part of the competition set for composition at some point, but not actually be composable (because of a failure of unification). This will result in the probability mass associated with that fragment being wasted. In order to avoid this probability leak, it is essential that every fragment in the competition set at a particular point actually be composable there. Thus, we must define competition sets relative to particular open slots, and include all fragments that can be successfully unified with the next open slot (NOS) of an active fragment. Since previous derivation steps can affect the specificity of such slots, this means that competition sets

cannot be predetermined. Suppose  $f_{i-1}$  is the structure produced before the  $i^{\text{th}}$  step of the derivation process. The probability associated with a fragment  $f_i$  is defined as in (15).<sup>8</sup>

$$(15) \quad P(f_i) = \frac{|f_i|}{\sum_{\substack{f \text{ is unifiable} \\ \text{with NOS}(f_{i-1})}} |f|}$$

### 3 Discussion and Concluding Remarks

We have presented the theoretical foundations of a DOP model which uses the formal and representational apparatus of HPSG. It is perhaps worth pointing out that this is a significant result in itself (it is not *a priori* obvious that a DOP model can be provided for any particular linguistic theory or formalism). Apart from the advantages that follow from the richer representational basis that such a model affords, it has a number of attractions. (i) It is able to produce fragments are at the right level of generality, so as to avoid both under- and over-generation, and avoid the combinatory explosion of fragments which accompanies use of a *Discard* operation; (ii) the probability model (relative frequency estimation) is very simple and easy to understand; (iii) the model is probabilistically well behaved, in the sense of defining a genuine probability distribution over fragments and representations. Notice that this means that it may in principle provide a general foundation for ‘probabilistic HPSG’.

There are a number of issues that deserve further attention.

We have assumed that in HPSG all grammatical phenomena can be captured in the type theory. This is nearly, but not quite, true. One area which seems to resist this is Binding Theory, where ‘Condition

<sup>8</sup>Fragment probabilities for the derivation initial fragments are more straightforward, and can e.g. be based on frequency estimation on *CAT* values.

Abney (1997) argues that relative frequency estimation is problematic for probabilistic Attribute-Value Grammars (AVGs) in general, and loglinear or maximum entropy models (Abney, 1997; Miyao and Tsujii, 2002) are generally considered to be more suitable for such formalisms. In this context, it is interesting that a relative frequency estimation approach can be used in connection with a rich formalism such as HPSG.

C' — the requirement that non-anaphoric NPs be unbound — resists this. See (Linardaki, 2006) for discussion and possible solutions.

From a practical, e.g. language engineering, point of view, a natural objection is this: one of the attractions of DOP is that it seems to dispense with the need to write grammars — all that is needed is a treebank. Clearly, this approach lacks this attraction, since it relies on the existence of a type system (i.e. an HPSG grammar). Several reactions are possible. One is that, from a theoretical point of view, it is reasonable to expect to have both a performance model (e.g. a form of DOP), and a (competence) grammar. Another is to note that in point of fact the existence of an HPSG treebank entails the existence of a type theory, however unarticulated or partial, in the representations themselves. Finally, one may note that from a practical point of view there is no need for the type-system to be a complete, precise or highly articulated grammar HPSG grammar in the normal sense — simply, the more precise it is, the more precise will be the results of type-expansion, and the less the system will overgenerate.

Another natural objection is that by ensuring that fragments have the 'right' level of generality, we have lost the robustness that is a feature of approaches that use a *Discard* operation (i.e. the ability to deal with input which is in some way ill-formed or extra-grammatical). One possibility would be to explore other approaches to robustness.<sup>9</sup> But of course one could also introduce a *Discard* operation directly: this would bring the advantages of robustness without the cost of a theoretically unsound characterization of ungrammaticality.

## References

- Steven P. Abney. 1997. Stochastic Attribute-Value Grammars. *Computational Linguistics*, 23(4):597–618.
- Rens Bod and Ronald Kaplan. 1998. A Probabilistic Corpus-Driven Model for Lexical Functional Analysis. In *Proceedings of COLING-ACL'98*, Montreal, Canada.
- Rens Bod and Ronald Kaplan. 2003. A DOP Model for Lexical-Functional Grammar. In Remko Scha Rens Bod and Khalil Sima'an, editors, *Data-Oriented Parsing*, chapter 12, pages 211–233. CSLI Publications, Stanford, California.
- Rens Bod, Remko Scha, and Khalil Sima'an, editors. 2003. *Data-Oriented Parsing*. CSLI Publications, Stanford, California.
- Rens Bod. 1995. *Enriching Linguistics with Statistics: Performance Models of Natural Language*. Ph.D. thesis, Universiteit van Amsterdam, Amsterdam, The Netherlands.
- Rens Bod. 2003. An Efficient Implementation of a New DOP Model. In *Proceedings of the EACL 2003*, pages 19–26, Budapest, Hungary, April.
- Frederik Fouvry. 2003. *Robust Processing for Constraint-based Grammar Formalisms*. Ph.D. thesis, University of Essex, April.
- Jonathan Ginzburg and Ivan A. Sag. 2000. *Interrogative Investigations*. CSLI Publications, Stanford.
- Evita Linardaki. 2006. *Linguistic and statistical extensions of Data Oriented Parsing*. Ph.D. thesis, University of Essex.
- Yusuke Miyao and Jun'ichi Tsujii. 2002. Maximum entropy estimation for feature forests. In *Proceedings of the Human Language Technology Conference*.
- Günter Neumann. 1999. Learning Stochastic Lexicalized Tree Grammars from HPSG. Technical report, DFKI, Saarbruecken.
- Günter Neumann. 2003. A Data-Driven Approach to Head-Driven Phrase Structure Grammar. In R. Scha R. Bod and K. Sima'an, editors, *Data-Oriented Parsing*, chapter 13, pages 233–251. CSLI Publications, Stanford.
- Carl J. Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago University Press.
- Derek Proudian and Carl J. Pollard. 1985. Parsing head-driven phrase structure grammar. In *Proceedings of the Twenty-Third Annual Meeting of the ACL*, pages 167–171, Chicago, IL. ACL.
- Gertjan van Noord. 1997. An efficient implementation of the head-corner parser. *Computational Linguistics*, 23(3):425–456.

<sup>9</sup>Robust unification (Fouvry, 2003), which is based on extending the signature to a lattice to include the unique joins of every set of incompatible types, is a promising alternative here.