2

A DOP model for Lexical-Functional Grammar

RENS BOD AND RONALD KAPLAN

2.1 Introduction

It is well-known that there exist many syntactic and semantic dependencies that are not reflected directly in a surface tree. All modern linguistic theories propose more articulated representations and mechanisms in order to characterize such linguistic phenomena. The Tree-DOP model is thus limited in that it cannot account for these phenomena. In this chapter, we show how a DOP model can be developed for the more articulated representations of Lexical-Functional Grammar (LFG), that is known to be beyond context-free (Kaplan & Bresnan 1982; Kaplan 1989). LFG representations consist of a surface constituent tree enriched with a corresponding functional structure. In order to develop a DOP model for LFG we will define new settings for the four parameters of the general DOP framework, i.e. (1) the representations, (2) the fragments, (3) the composition operation, and (4) the probability model.

We will see that the resulting LFG-DOP model triggers a new, corpus-based notion of grammaticality, and an interestingly different class of probability models. We present a parser which uses fragments from LFG-

annotated corpora to analyze new sentences, and Monte Carlo techniques to estimate the most probable analysis. We test some versions of the model on the Verbmobil and Homecentre corpora, showing among other things that the parse accuracy increases with increasing fragment size (as with Tree-DOP), and that LFG-DOP outperforms Tree-DOP if evaluated on tree structures.

2.2 Extending Tree-DOP to Lexical-Functional Grammar Representations

2.2.1 Representations

The representations used by LFG-DOP are directly taken from LFG theory, that is, every utterance is annotated with a c-structure, an f-structure and a mapping ϕ between them. The c-structure is a tree that describes the surface constituent structure of an utterance; the f-structure is an attribute-value matrix marking such grammatical relations as subject, predicate and object, as well as providing agreement features and semantic forms; and ϕ is a correspondence function that maps nodes of the c-structure into units of the f-structure (Kaplan & Bresnan 1982; Kaplan 1989). The following figure shows a representation for the utterance *Kim eats*. (We leave out some features to keep the example simple.)

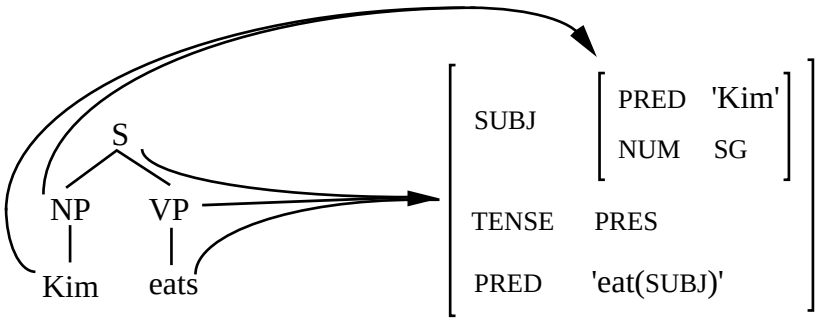

Figure 1. An LFG representation for *Kim eats*

Note that the ϕ correspondence function gives an explicit characterization of the relation between the superficial and underlying syntactic properties of an utterance, indicating how certain parts of the string carry information about particular units of underlying structure. As such, it will play a crucial role in our definition for the decomposition and composition operations of LFG-

DOP. In figure 1 we see for instance that the NP node maps to the subject f-structure, and the S and VP nodes map to the outermost f-structure.

It is generally the case that the nodes in a subtree carry information only about the f-structure units that the subtree's root gives access to. The notion of accessibility is made precise in the following definition:

> An f-structure unit *f* is ϕ-*accessible* from a node *n* iff either *n* is ϕ-linked to *f* (that is, $f = \phi(n)$) or *f* is contained within $\phi(n)$ (that is, there is a chain of attributes that leads from $\phi(n)$ to *f*).

All the f-structure units in figure 1 are ϕ-accessible from for instance the S node and the VP node, but the TENSE and top-level PRED are not ϕ-accessible from the NP node.

According to LFG theory, c-structures and f-structures must satisfy certain formal well-formedness conditions. A c-structure/f-structure pair is a *valid* LFG representation only if it satisfies the Nonbranching Dominance, Uniqueness, Coherence and Completeness conditions (Kaplan & Bresnan 1982). Nonbranching Dominance demands that no c-structure category appears twice in a nonbranching dominance chain; Uniqueness asserts that there can be at most one value for any attribute in the f-structure; Coherence prohibits the appearance of grammatical functions that are not governed by the lexical predicate; and Completeness requires that all the functions that a predicate governs appear as attributes in the local f-structure. The first three conditions (Nonbranching Dominance, Uniqueness and Coherence) are monotonic, in the sense that if they are unsatisfied by a substructure they will also be unsatisfied by any superstructure. The Completeness condition, on the other hand, is non-monotonic in that larger structures may satisfy this condition while their substructures do not (see Kaplan & Bresnan 1982).

2.2.2 Fragments

Many different DOP models are compatible with the system of LFG representations. Following Bod and Kaplan (1998), we give a relatively straightforward extension of Tree-DOP where the fragments are extended to take correspondences and f-structure features into account. That is, the fragments for LFG-DOP consist of connected subtrees whose nodes are in ϕ-correspondence with the correponding sub-units of f-structures. To give a precise definition of LFG-DOP fragments, it is convenient to redefine the

fragments of Tree-DOP (chapter 1) in terms of fragment-producing operations, which we will call *decomposition operations*.

The fragments of Tree-DOP can be defined by the following two decomposition operations:

(1) *Root*: the *Root* operation selects any node of a tree to be the root of the new subtree and erases all nodes except the selected node and the nodes it dominates.

(2) *Frontier*: the *Frontier* operation then chooses a set (possibly empty) of nodes in the new subtree different from its root and erases all subtrees dominated by the chosen nodes.

Notice that *Root* and *Frontier* define exactly the same bag of subtrees as the fragment definitions (1)-(3) in chapter 1. We now extend *Root* and *Frontier* so that they also apply to the nodes of the c-structure in LFG, while respecting the fundamental principles of c-structure/f-structure correspondence.

When a node is selected by the *Root* operation, all nodes outside of that node's subtree are erased, just as in Tree-DOP. Further, for LFG-DOP, all ϕ links leaving the erased nodes are removed and all f-structure units that are not ϕ-accessible from the remaining nodes are erased. *Root* thus maintains the intuitive correlation between nodes and the information in their corresponding f-structures. For example, if *Root* selects the NP in figure 1, then the f-structure corresponding to the S node is erased, giving figure 2 as a possible fragment:

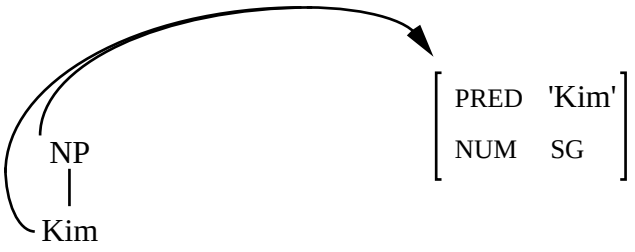

Figure 2. A fragment obtained by the *Root* operation

In addition the *Root* operation deletes from the remaining f-structure all semantic forms that are local to f-structures that correspond to erased c-structure nodes, and it thereby also maintains the fundamental two-way

connection between words and meanings. Thus, if *Root* selects the VP node so that the NP is erased, the subject semantic form "Kim" is also deleted:

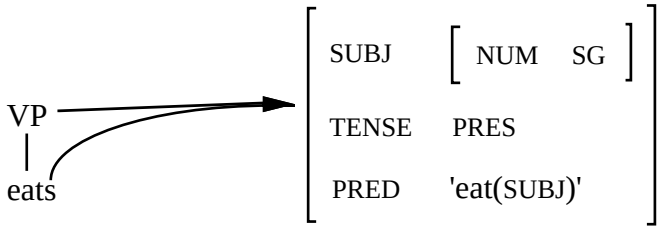

Figure 3. Another *Root*-generated fragment

As with Tree-DOP, the *Frontier* operation then selects a set of frontier nodes and deletes all subtrees they dominate. Like *Root*, it also removes the ϕ links of the deleted nodes and erases any semantic form that corresponds to any of those nodes. *Frontier* does not delete any other f-structure features. This reflects the fact that all features are ϕ-accessible from the fragment's root even when nodes below the frontier are erased. For instance, if the VP in figure 1 is selected as a frontier node, *Frontier* erases the predicate "eat(SUBJ)" from the fragment:

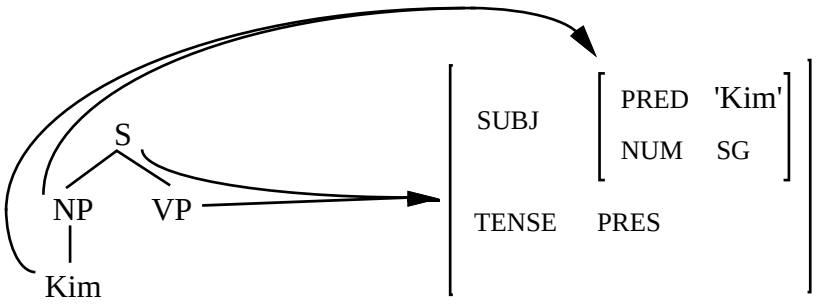

Figure 4. A fragment obtained by the *Frontier* operation

Note that the *Root* and *Frontier* operations retain the subject's NUM feature in the VP-rooted fragment of figure 3, even though the subject NP is not present. This reflects the fact, usually encoded in particular grammar rules or lexical entries, that verbs of English carry agreement features for their subjects. On the other hand, the fragment in figure 4 retains the predicate's TENSE feature, reflecting the possibility that English subjects might also

carry information about their predicate's tense. Subject-tense agreement as encoded in figure 4 is a pattern seen in some languages (e.g. the split-ergativity pattern of languages like Hindi, Urdu and Georgian) and thus there is no universal principle by which fragments such as in figure 4 can be ruled out. But in order to represent directly the possibility that subject-tense agreement is not a dependency of English, we also allow an S fragment in which the TENSE feature is deleted, as in figure 5.

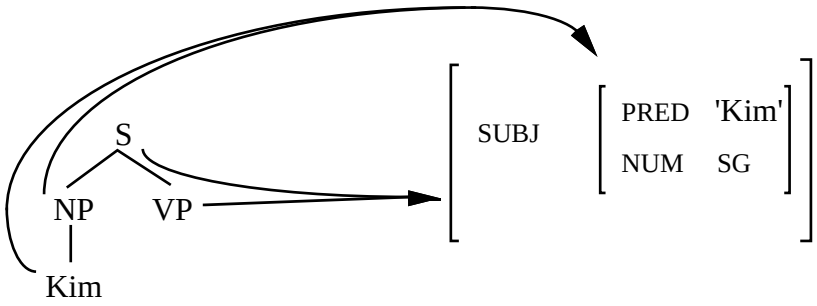

Figure 5. A fragment obtained by the *Discard* operation

The fragment in figure 5 is produced by a third decomposition operation, *Discard*, defined to construct generalizations of the fragments supplied by *Root* and *Frontier*. *Discard* acts to delete combinations of attribute-value pairs subject to the following restriction: *Discard* does not delete pairs whose values ϕ-correspond to remaining c-structure nodes.

This condition maintains the essential correspondences of LFG representations: if a c-structure and an f-structure are paired in one fragment provided by *Root* and *Frontier*, then *Discard* also pairs that c-structure with all generalizations of that fragment's f-structure. For convenience, we will sometimes use the term *generalized* fragment to indicate a fragment generated by one or more applications of the *Discard* operation. The fragment in figure 5 results from applying *Discard* to the TENSE feature in figure 4. *Discard* also produces fragments such as figure 6, where the subject's number in figure 3 has been deleted:

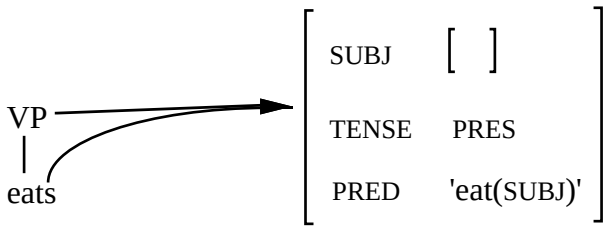

Figure 6. Another fragment obtained by the *Discard* operation

Again, since we have no language-specific knowledge apart from the corpus, we have no basis for ruling out fragments like figure 6. Indeed, it is quite intuitive to omit the subject's number in fragments derived from sentences with past-tense verbs or modals. Thus the specification of *Discard* reflects the fact that LFG representations, unlike LFG grammars, do not indicate unambiguously the c-structure source (or sources) of their f-structure feature values.

2.2.3 The composition operation

In LFG-DOP the operation for combining fragments, again indicated by ∘, is carried out in two steps. First the c-structures are combined by left-most substitution subject to the category-matching condition, just as in Tree-DOP. This is followed by the recursive unification of the f-structures corresponding to the matching nodes. The result retains the ϕ correspondences of the fragments being combined. A derivation for an LFG-DOP representation *R* is a sequence of fragments the first of which is labeled with S and for which the iterative application of the composition operation produces *R*.

We illustrate the two-stage composition operation by means of a simple example. We therefore assume a corpus containing the representation in figure 1 for the sentence *Kim eats* and the representation in figure 7 for the sentence *John fell*.

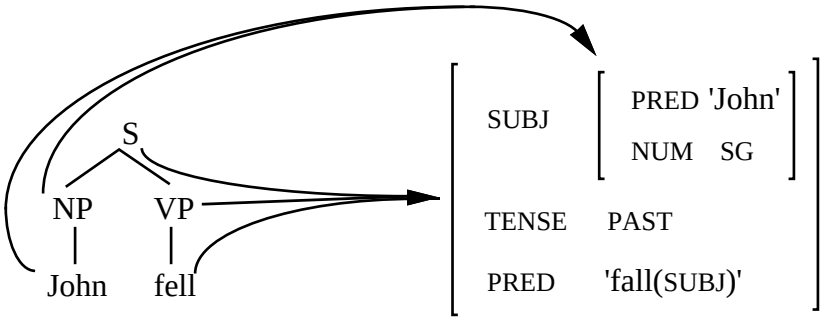

Figure 7. A representation for *John fell*

Figure 8 shows the effect of the LFG-DOP composition operation using two fragments from this corpus. The NP-rooted fragment is substituted for the NP in the first fragment, and the second f-structure unifies with the first f-structure, resulting into a representation for the new sentence *Kim fell*.

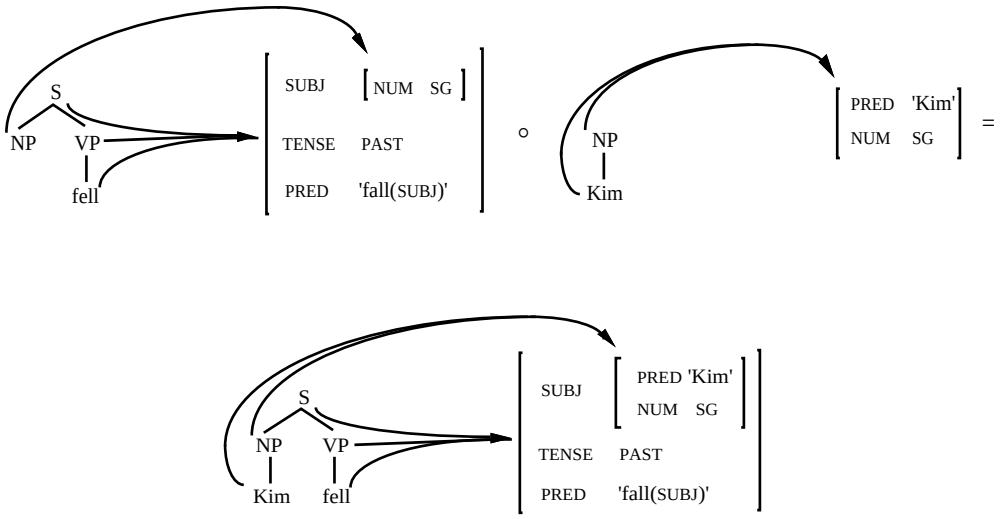

Figure 8. Illustration of the LFG-DOP composition operation

This representation satisfies the well-formedness conditions and is therefore valid. Note that in LFG-DOP, as in the tree-based DOP models, the same representation may be produced by several distinct derivations involving different fragments.

While the example sentence *Kim fell* is clearly grammatical, LFG-DOP can also produce representations for sentences that are intuitively ungrammatical. To show this, we extend our example corpus with the representation in figure 9 for the sentence *People ate*.

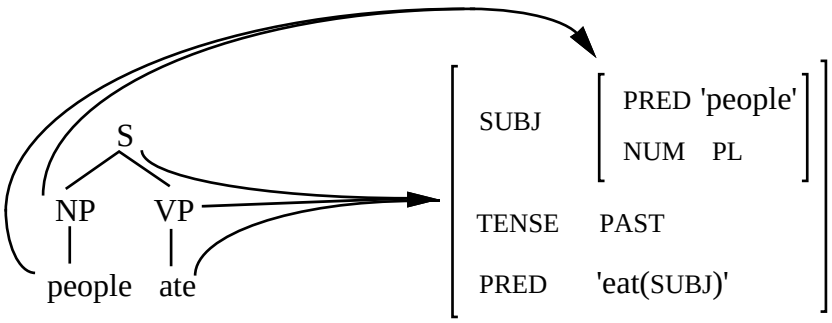

Figure 9. A representation for *People ate*

Then the following derivation produces a valid representation for the intuitively ungrammatical sentence *People eats* (where the second fragment is produced by discarding the number feature of *eats*):

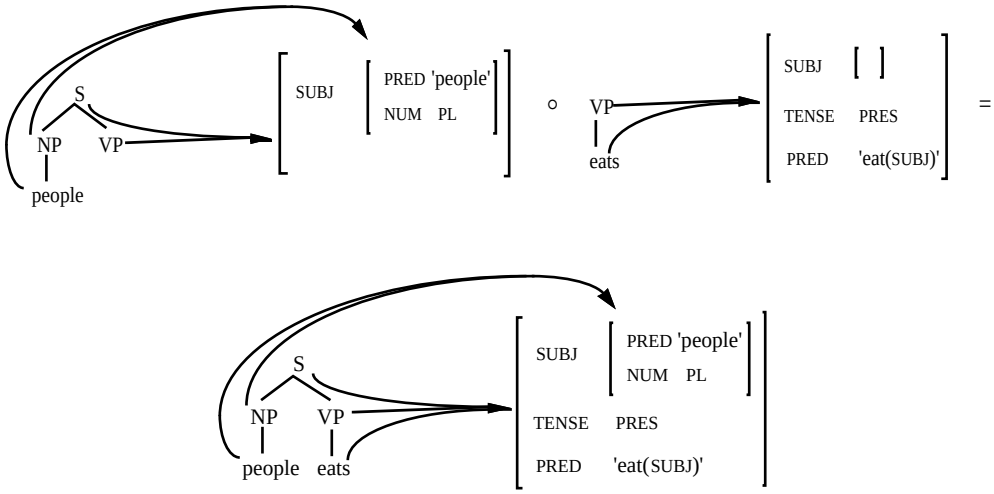

Figure 10. A valid representation for an intuitively ungrammatical sentence

Thus this representation assigns a *plural* interpretation to the sentence *People eats*. Note that LFG-DOP can also produce a (valid) representation which

assigns a *singular* interpretation to *People eats*, if the number feature of *people* rather than *eats* is discarded. Finally, LFG-DOP produces a (valid) representation with an *unmarked* number value if the number features of both *people* and *eats* are discarded. (It is left to the probability model which of these representations is ranked highest.)

This system of fragments and composition thus provides a representational basis for a robust model of language comprehension in that it assigns at least some representations to many strings that would generally be regarded as ill-formed. A correlate of this advantage, however, is the fact it does not offer a direct formal account of metalinguistic judgments of grammaticality. Nevertheless, we can reconstruct the notion of grammaticality by means of the following definition:

> A sentence is *grammatical with respect to a corpus* if and only if it has at least one valid representation with at least one derivation without generalized fragments.

Thus the system is robust in that it assigns three representations (singular, plural, and unmarked as the subject's number) to the string *People eats*, based on fragments for which the number feature of *people*, *eats*, or both has been discarded. But unless the corpus contains non-plural instances of *people* or non-singular instances of *eats*, there will be no *Discard*-free derivation and the string will be classified as ungrammatical (with respect to the corpus).

2.2.4 Probability models

As in Tree-DOP, an LFG-DOP representation *R* can typically be derived in many different ways. Thus, if each derivation *D* has a probability P(*D*), then the probability of deriving *R* is again the sum of the individual derivation probabilities:

$$(1) \qquad P(R) = \sum_{D \text{ derives } R} P(D)$$

An LFG-DOP derivation is produced by a stochastic process which starts by randomly choosing a fragment whose c-structure is labeled with the initial category (e.g. S). At each subsequent step, a next fragment is chosen at random from among the fragments that can be composed with the current subanalysis. The chosen fragment is composed with the current subanalysis to produce a new one; the process stops when an analysis results with no non-terminal leaves. We call the set of composable fragments at a certain step

in the stochastic process the competition set at that step. Let CP(f | CS) denote the probability of choosing a fragment f from a competition set CS containing f, then the probability of a derivation $D = \langle f_1, f_2 \ldots f_k \rangle$ is

$$(2) \qquad P(\langle f_1, f_2 \ldots f_k \rangle) = \prod_i CP(f_i \mid CS_i)$$

where the *competition probability* CP(f | CS) is expressed in terms of fragment probabilities P(f):

$$(3) \qquad CP(f \mid CS) = \frac{P(f)}{\sum_{f' \in CS} P(f')}$$

Tree-DOP is the special case where there are no conditions of validity other than the ones that are enforced on-line at each step of the stochastic process by the composition operation. This is not generally the case and is certainly not the case for the Completeness Condition of LFG representations: Completeness is a property of a final representation that cannot be evaluated at any intermediate steps of the process (we will return to this property below). However, we can define probabilities for the valid representations by sampling only from such representations in the output of the stochastic process. The probability of sampling a particular valid representation R is given by

$$(4) \qquad P(R \mid R \text{ is valid}) = \frac{P(R)}{\sum_{R' \text{ is valid}} P(R')}$$

This formula assigns probabilities to valid representations whether or not the stochastic process guarantees validity. The valid represention for a particular word string W are obtained by a further sampling step and their probabilities are given by:

$$(5) \qquad P(R \mid R \text{ is valid and yields } W) = \frac{P(R)}{\sum_{R' \text{ is valid and yields } W} P(R')}$$

The formulas (1) through (5) will be part of any LFG-DOP probability model. The models will differ only in how the competition sets are defined, and this in turn depends on which well-formedness conditions are enforced

on-line during the stochastic branching process and which are evaluated by the off-line validity sampling process.

One model, which we call M1, is a straightforward extension of Tree-DOP's probability model. This computes the competition sets only on the basis of the category-matching condition, leaving all other well-formedness conditions for off-line sampling. Thus for M1 the competition sets are defined simply in terms of the categories of a fragment's c-structure root node. Suppose that $F_{i-1} = f_1 \circ f_2 \circ ... \circ f_{i-1}$ is the current subanalysis at the beginning of step i in the process, that LNC(F_{i-1}) denotes the category of the leftmost nonterminal node of the c-structure of F_{i-1}, and that r(f) is interpreted as the root node category of f's c-structure component. Then the competition set for the i^{th} step is

(6) $CS_i = \{ f : r(f) = LNC(F_{i-1}) \}$

Since these competition sets depend only on the category of the leftmost nonterminal of the current c-structure, the competition sets group together all fragments with the same root category, independent of any other properties they may have or that a particular derivation may have. The competition probability for a fragment can be expressed by the formula

(7) $$CP(f) = \frac{P(f)}{\sum_{f' : r(f')=r(f)} P(f')}$$

We see that the choice of a fragment at a particular step in the stochastic process depends only on the category of its root node; other well-formedness properties of the representation are not used in making fragment selections. Thus, with this model the stochastic process may produce many invalid representations; we rely on sampling of valid representations and the conditional probabilities given by (4) and (5) to take the Uniqueness, Coherence, and Completeness Conditions into account.

Another possible model (M2) defines the competition sets so that they take a second condition, Uniqueness, into account in addition to the root node category. For M2 the competing fragments at a particular step in the stochastic derivation process are those whose c-structures have the same root node category as LNC(F_{i-1}) and also whose f-structures are consistently unifiable with the f-structure of F_{i-1}. Thus the competition set for the i^{th} step is

(8) $CS_i = \{ f : r(f) = LNC(F_{i-1}) \text{ and } f \text{ is unifiable with f-structure of } F_{i-1}\}$

Although it is still the case that the category-matching condition is independent of the derivation, the unifiability requirement means that the competition sets vary according to the representation produced by the sequence of previous steps in the stochastic process. Unifiability must be determined at each step in the process to produce a new competition set, and the competition probability remains dependent on the particular step:

$$(9) \qquad CP(f_i \mid CS_i) = \frac{P(f_i)}{\sum_{f:\, r(f)=r(f_i) \text{ and } f \text{ is unifiable with } F_{i-1}} P(f)}$$

On this model we again rely on sampling and the conditional probabilities (4) and (5) to take just the Coherence and Completeness Conditions into account.

In model M3 we define the stochastic process to enforce three conditions, Coherence, Uniqueness and category-matching, so that it only produces representations with well-formed c-structures that correspond to coherent and consistent f-structures. The competition probabilities for this model are given by the obvious extension of (9). It is not possible, however, to construct a model in which the Completeness Condition is enforced during the derivation process. This is because the satisfiability of the Completeness Condition depends not only on the results of previous steps of a derivation but also on the following steps (see Kaplan and Bresnan 1982). This nonmonotonic property means that the appropriate step-wise competition sets cannot be defined and that this condition can only be enforced at the final stage of validity sampling.

In each of these three models the category-matching condition is evaluated on-line during the derivation process while other conditions are either evaluated on-line or off-line by the after-the-fact sampling process. LFG-DOP is crucially different from the tree-based DOP models in that at least one validity requirement, the Completeness Condition, must always be left to the post-derivation process. Note that a number of other models are possible which enforce other combinations of these three conditions. However, in our experiments in section 2.4 we will only test model M3, as this model selects only those fragments at each derivation step that may result in a valid LFG representation, thus reducing the off-line validity checking just to the Completeness condition.

Note that the computation of the competition probability in the above formulas still requires a definition for the fragment probability P(*f*). In Bod and Kaplan (1998), we defined the probability of a fragment simply as its relative frequency in the bag of all fragments generated from the corpus, just as in most Tree-DOP models. We will refer to this fragment estimator as "simple relative frequency" or "simple RF". The simple RF estimator does not distinguish between *Root*/*Frontier*-generated fragments and *Discard*-generated fragments, the latter being in fact generalizations over *Root*/*Frontier*-generated fragments. Although we showed in Bod and Kaplan (1998) with an example that the simple RF estimator exhibits a preference for the most specific representation containing the fewest feature generalizations (mainly because specific representations tend to have more derivations than generalized representations), we did not perform any empirical evaluation. In this paper, we will assess the simple RF estimator in section 2.4.

However, we will also assess an alternative definition of fragment probability which is a refinement of simple RF. This alternative fragment probability definition *does* distinguish between fragments supplied by *Root*/*Frontier* and fragments supplied by *Discard*. We will treat the first type of fragments as seen events, and the second type of fragments as previously unseen events. We thus create two separate bags corresponding to two separate distributions: a bag with fragments generated by *Root* and *Frontier*, and a bag with fragments generated by *Discard*. We assign probability mass to the fragments of each bag by means of *discounting*: the relative frequencies of seen events are discounted and the gained probability mass is reserved for the bag of unseen events (cf. Ney et al. 1997). We accomplish this by a very simple estimator: the Turing-Good estimator (Good 1953) which computes the probability mass of unseen events as n_1/N where n_1 is the number of singleton events and N is the total number of seen events. This probability mass is assigned to the bag of *Discard*-generated fragments. The remaining mass $(1 - n_1/N)$ is assigned to the bag of *Root*/*Frontier*-generated fragments. Thus the total probability mass is redistributed over the seen and unseen fragments. The probability of each fragment is then computed as its relative frequency in its bag multiplied by the probability mass assigned to this bag. Let $|f|$ denote the frequency of a fragment *f*, then its probability is given by:

(10) $P(f \mid f \text{ is generated by } Root/Frontier) = (1 - n_1/N)\frac{|f|}{\sum_{f'\text{: } f' \text{ is generated by } Root/Frontier} |f'|}$

(11) $P(f \mid f \text{ is generated by } Discard) = (n_1/N)\frac{|f|}{\sum_{f'\text{: } f' \text{ is generated by } Discard} |f'|}$

We will refer to this fragment probability estimator as "discounted relative frequency" or "discounted RF". Note that the discounted RF estimator assigns less probability mass to *Discard*-generated fragments than the simple RF estimator. For each *Root*/*Frontier*-generated fragment there are exponentially many *Discard*-generated fragments (exponential in the number of features the fragment contains), which means that the *Discard*-generated fragments absorb a vast amount of probability mass under the simple RF estimator. The discounted RF estimator, on the other hand, assigns a fixed probability mass to the distribution of *Discard*-generated fragments and therefore the exponential explosion of these fragments does not affect the probabilities of *Root*/*Frontier*-generated fragments. We want to note that neither of the two relative frequency estimators maximizes the likelihood of the training data (cf. Abney 1997). The application of log-linear or maximum entropy models to LFG-DOP (Berger et al. 1996) will be explored in the future.[1]

2.3 Parsing with LFG-DOP

In his PhD-thesis, Cormons (1999: 71-96) describes a parsing algorithm for LFG-DOP which is based on the Tree-DOP parsing technique described in Bod (1998: 40-50). Cormons first converts LFG-representations into more compact indexed trees: each node in the c-structure is assigned an index which refers to the ϕ-corresponding f-structure unit. For example, the representation in figure 7 is indexed as

[1] The reason to do this future research is not to meet some particular requirement of statistical theory but to determine what kind of estimator is the true one, i.e. the one that the psychological system (whose interpretation judgments we are trying to account for) is using.

(S.1 (NP.2 John.2)
(VP.1 fell.1))

where

1 --> [(SUBJ = 2)
(TENSE = PAST)
(PRED = fall(SUBJ))]

2 --> [(PRED = John)
(NUM = SG)]

The indexed trees are then fragmented by Tree-DOP as described in section 2.2. Next, the LFG-DOP decomposition operations *Root*, *Frontier* and *Discard* are applied to the f-structure units that correspond to the indices in the c-structure subtrees. Having obtained the set of LFG-DOP fragments in this way, each test sentence is parsed by a bottom-up chart parser using initially the indexed subtrees only. As shown in Bod (1993, 1995), standard chart parsing techniques can be used by converting subtrees into rewrite rules.

Thus only the Category-matching condition is enforced during the chart-parsing process. The Uniqueness and Coherence conditions of the corresponding f-structure units are enforced during the disambiguation (or chart-decoding) process. Disambiguation is accomplished by computing a large number of random derivations from the chart and by selecting the analysis which results most often from these derivations ("Monte Carlo disambiguation", see part II). In LFG-DOP, sampling a random derivation from the chart consists of choosing at random one of the fragments from the set of *composable* fragments at every labeled chart-entry (where the random choices at each chart-entry are based on the probabilities of the fragments). The derivations are sampled in a top-down, leftmost order so as to maintain the LFG-DOP derivation order. Thus the competition sets of composable fragments are computed on the fly during the Monte Carlo sampling process by grouping the f-structure units that unify and that are coherent with the subderivation built so far.

As mentioned in sections 2.2.1 and 2.2.4, the Completeness condition can only be checked after the derivation process. Incomplete derivations are simply removed from the sampling distribution. After sampling a sufficiently large number of random derivations that satisfy the LFG validity requirements, the most probable analysis is estimated by the analysis which results most often from the sampled derivations. As a stop condition on the

number of sampled derivations, we compute with intervals of 100 samples the probability of error; this is the probability that the analysis which is most frequently generated by the sampled derivations is not equal to the most probable analysis (see Bod 1998: 45-50). We set this error probability to 0.05 in our experiments. However, if there is no unique most probable analysis, the sampling process will of course not converge on one outcome. In order to rule out the possibility that the sampling process would never stop, we enforce a maximum sample size of10,000 derivations.

2.4 Experiments with LFG-DOP

We performed some experiments with LFG-DOP with two LFG-annotated corpora: the Verbmobil corpus and the Homecentre corpus. Both corpora were annotated at Xerox PARC. They contain packed LFG-representations (Maxwell & Kaplan 1991) of the grammatical parses (c-structures and f-structures) of each sentence, together with an indication which of these parses is the correct one. For our experiments we only used the correct parse of each sentence resulting in 540 Verbmobil parses and 980 Homecentre parses. Each corpus was divided into a 90% training set and a 10% test set. This division was random except for one constraint: that all the words in the test set actually occurred in the training set. The sentences from the test set were parsed and disambiguated by means of the fragments from the training set. Due to memory limitations, we restricted the maximum depth of the indexed subtrees to 4. Because of the small size of the corpora we averaged our results on 10 different training/test set splits. Besides an *exact match* accuracy metric, we also used somewhat less-strict measures based on the well-known PARSEVAL metrics that evaluate phrase-structure trees (Black et al. 1991). The PARSEVAL metrics compare a proposed parse *P* with the corresponding correct treebank parse *T* as follows:

$$\text{Precision} = \frac{\#\text{ correct constituents in } P}{\#\text{ constituents in } P}$$

$$\text{Recall} = \frac{\#\text{ correct constituents in } P}{\#\text{ constituents in } T}$$

According to the original PARSEVAL scheme, a constituent in *P* is "correct" if there exists a constituent in *T* of the same label that spans the same words. It

is not obvious how to extend this tree-based scheme to LFG representations. In this paper we have picked one which we think understates our accuracy, perhaps significantly, but it is an extremely simple extension of the PARSEVAL scheme. We will explore other measures in future work. We thus extend the PARSEVAL notion of "correct constituent" in the following way: a constituent in *P* is correct if there exists a constituent in *T* of the same label that spans exactly the same words and that ϕ-corresponds to the same f-structure unit.

2.4.1 Comparing the two fragment estimators

We were first interested in comparing the performance of the simple RF estimator, which treats all fragments probabilistically equally, against the discounted RF estimator, which distinguishes between generalized and ungeneralized fragments. Furthermore, we want to study the contribution of generalized fragments to the parse accuracy. We therefore created for each training set two sets of fragments: one which contains *all* fragments (up to depth 4) and one which excludes the generalized fragments as generated by *Discard*. The exclusion of these *Discard*-generated fragments means that all probability mass goes to the fragments generated by *Root* and *Frontier* in which case the two estimators are equivalent. The following two tables present the results of our experiments where +Discard refers to the full set of fragments and –Discard refers to the fragment set without *Discard*-generated fragments.

Estimator	Exact Match		Precision		Recall	
	+Discard	–Discard	+Discard	–Discard	+Discard	–Discard
Simple RF	1.1%	35.2%	13.8%	76.0%	11.5%	74.9%
Discounted RF	35.9%	35.2%	77.5%	76.0%	76.4%	74.9%

Table 1. Experimental results on Verbmobil for fragment-depth ≤ 4

Estimator	Exact Match		Precision		Recall	
	+Discard	−Discard	+Discard	−Discard	+Discard	−Discard
Simple RF	2.7%	37.9%	17.1%	77.8%	15.5%	77.2%
Discounted RF	38.4%	37.9%	80.0%	77.8%	78.6%	77.2%

Table 2. Experimental results on Homecentre for fragment-depth ≤ 4

The tables show that the simple RF estimator scores extremely badly if all fragments are used: the exact match is only 1.1% on the Verbmobil corpus and 2.7% on the Homecentre corpus, whereas the discounted RF estimator scores respectively 35.9% and 38.4% on these corpora. Also the precision and recall scores obtained with the simple RF estimator are quite low: e.g. 13.8% and 11.5% on the Verbmobil corpus, where the discounted RF estimator obtains 77.5% and 76.4%. We found that even for the few test sentences that occur literally in the training set, the simple RF estimator does not always generate the correct analysis, whereas the discounted RF estimator does. Interestingly, the accuracy of the simple RF estimator is much higher if *Discard*-generated fragments are excluded. This suggests that treating generalized fragments probabilistically in the same way as ungeneralized fragments is harmful. Cormons (1999: 64) made a mathematical observation which also shows that generalized fragments can get too much probability mass under the simple RF estimator, leading to biased predictions for the best parse. Thus, generalized fragments should really be seen as "previously unobserved fragments" whose probability should be estimated by discounting.

The tables also show that the inclusion of *Discard*-generated fragments leads only to a slight accuracy increase under the discounted RF estimator. According to paired *t*-testing, only the differences in precision scores on the Homecentre corpus were statistically significant. Thus except for one metric on one corpus, *Discard*-generated fragments do not significantly contribute to the parse accuracy on these corpora. Of course, these generalized fragments remain important for parsing sentences which are "ungrammatical with respect to the corpus", which was the original motivation for including them.

To put our results in another perspective, we calculated the parse accuracy by randomly picking a parse from the derivation forest for each test

sentence without taking into account the fragment probabilities. This resulted in an exact match of 0% for both corpora and for all training/test set splits. Interestingly, the difference between the 0% accuracy and the 1.1% accuracy obtained with simple RF on the Verbmobil corpus was statistically insignificant (though the difference was significant for the Homecentre corpus). Thus for Verbmobil sentences, the use of simple RF as a fragment estimator does not perform significantly better than picking a parse by chance.

2.4.2 Comparing different fragment sizes

Next, we were interested in testing whether the general DOP hypothesis in Bod (1998) (which states that parse accuracy increases with increasing fragment size) can confirmed for LFG representations. We therefore performed a series of experiments where the fragment set is restricted to fragments of a certain maximum size. We defined the size of a fragment by its depth, which is the longest path from root to leaf of the fragment's c-structure unit. We used the same training/test set splits as in the previous experiments and used both ungeneralized and generalized fragments together with the discounted RF estimator. The following tables show the results for four different maximum fragment depths.

Fragment Depth	Exact Match	Precision	Recall
1	30.6%	74.2%	72.2%
≤2	34.1%	76.2%	74.5%
≤3	35.6%	76.8%	75.9%
≤4	35.9%	77.5%	76.4%

Table 3. Accuracies on the Verbmobil corpus for different fragment depths

Fragment Depth	Exact Match	Precision	Recall
1	31.3%	75.0%	71.5%
≤2	36.3%	77.1%	74.7%
≤3	37.8%	77.8%	76.1%
≤4	38.4%	80.0%	78.6%

Table 4. Accuracies on the Homecentre for different fragment depths

The tables show that there is an increase in accuracy if larger fragments are included. This result is significant in that it extends the plausibility of the DOP hypothesis to the linguistically sophisticated LFG representations (keeping in mind that our results were obtained on relatively small corpora). According to paired *t*-testing, all differences between the minimal and maximal accuracies for each metric are statistically significant. Note that our result is not self-evident: since LFG representations contain much more linguistic information than simple tree representations, it could have been the case that maximal accuracy was already achieved by a minimal set of depth-1 fragments (especially since minimal fragments already contain grammatical features and semantic forms). Yet, our experiments show that there is a significant increase in parse accuracy if counts of larger fragments are included.

2.4.3 Comparing LFG-DOP to Tree-DOP

Finally, we were interested in the impact of functional structures on predicting the correct tree structures. We therefore removed all f-structure units from the fragments, thus yielding a Tree-DOP model, and compared the results against the full LFG-DOP model (using the discounted RF estimator and all fragments up to depth 4). We evaluated the parse accuracy on the tree structures only, using exact match together with the standard PARSEVAL measures. We used the same training/test set splits as in the previous experiments. The following tables show the results.

Model	Exact Match	Precision	Recall
Tree-DOP	46.6%	88.9%	86.7%
LFG-DOP	50.8%	90.3%	88.4%

Table 5. Tree structure accuracy on the Verbmobil corpus

Model	Exact Match	Precision	Recall
Tree-DOP	49.0%	93.4%	92.1%
LFG-DOP	53.2%	95.8%	94.7%

Table 6. Tree structure accuracy on the Homecentre corpus

The results indicate that LFG-DOP's functional structures help to improve the parse accuracy of tree structures. In other words, LFG-DOP outperforms Tree-DOP if evaluated on tree structures only. According to paired *t*-testing all differences in accuracy were statistically significant. This result is promising since Bod (2001) reports that Tree-DOP obtains higher performance on the Wall Street Journal corpus than other models (e.g. Collins 2000; Charniak 2000). This suggests that LFG-DOP may even further improve the parse accuracy on the Wall Street Journal provided that the functional annotations in the Penn Treebank (Marcus et al. 1994) can be converted into LFG-style functional structures.

2.5 Conclusion

We have developed and tested a Data-Oriented Parsing model based on the syntactic representations of Lexical-Functional Grammar theory: LFG-DOP. We have seen that LFG-DOP triggers a new, corpus-based notion of grammaticality, and an interestingly different class of probability models. We described a parser which analyzes new input by combining fragments from LFG-annotated corpora into new analyses, and which uses Monte Carlo techniques to estimate the most probable analysis. We proposed and tested

two fragment estimators, one based on simple relative frequency and one based on discounted relative frequency. Our experiments showed that the discounted relative frequency estimator outperforms the simple relative frequency estimator.

The most significant experimental result in this paper, we believe, is the extension of the DOP hypothesis to LFG representations. We argued that this result is not self-evident: since LFG representations contain much more linguistic information than simple tree representations, it could have been the case that maximal accuracy was already achieved by a minimal set of depth-1 fragments. Yet, our experiments show that there is a significant increase in parse accuracy if counts of larger fragments are included.

Finally, we showed that LFG's functional structures contribute to significantly higher parse accuracy on tree structures, thus outperforming Tree-DOP. This suggests that our model may be successfully used to exploit the functional annotations in the Penn Treebank (Marcus et al. 1994), provided that these annotations can be converted into LFG-style functional structures. We must keep in mind that the results in this paper were obtained on the relatively small Verbmobil and Homecentre corpora (especially if compared with Wall Street Journal corpus on which Tree-DOP was tested, which contains over 40,000 sentences). One of the main goals for the future is to develop larger LFG-annotated corpora and to test LFG-DOP on these corpora. Another future goal is to test LFG-DOP under different probability models, such as log-linear or maximum entropy models (Abney 1997; Riezler et al. 2000) that maximize that likelihood of the training data. We also intend to find linguistic constraints on the *Discard* operation, a direction which is suggested by Way (1999).

Acknowledgements

This paper is the continuation of work begun in collaboration with Remko Scha and Khalil Sima'an. The initial stages of this work were carried out while the second author was a fellow of the Netherlands Institute for Advanced Study (NIAS). Subsequent stages were also carried out while the first author was a consultant at Xerox PARC. Boris Cormons contributed to our understanding of the probability models and implemented the first LFG-DOP parser which formed the basis for the parser used in this paper. Hadar Shemtov provided us with the relevant software for decoding the packed LFG-representations. Chris Manning and Andy Way contributed to our understanding of the properties of the *Discard* operation. We also benefitted

from our interactions with Joan Bresnan, Mary Dalrymple, Mark Johnson, Martin Kay, John Maxwell, Stanley Peters, Stefan Riezler, Remko Scha and Khalil Sima'an.

References

S. Abney, 1997. "Stochastic Attribute-Value Grammars", *Computational Linguistics*, 23(4), 597-617.

A. Berger, V. della Pietra and S. della Pietra, 1996. "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*, 22(1), 39-71.

E. Black et al., 1991. "A Procedure for Quantitatively Comparing the Syntactic Coverage of English", *Proceedings DARPA Speech and Natural Language Workshop*, Pacific Grove, Morgan Kaufmann.

R. Bod, 1993. "Using an Annotated Language Corpus as a Virtual Stochastic Grammar", *Proceedings AAAI'93*, Washington D.C.

R. Bod, 1995. *Enriching Linguistics with Statistics: Performance Models of Natural Language*, ILLC Dissertation Series 1995-14, University of Amsterdam, The Netherlands.

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

R. Bod, 2001. "What is the Minimal Set of Fragments which Obtains Maximum Parse Accuracy?", *Proceedings ACL'2001*, Toulouse, France.

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

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

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

B. Cormons, 1999. *Analyse et désambiguisation: Une approche à base de corpus (Data-Oriented Parsing) pour les répresentations lexicales fonctionnelles*. PhD thesis, Université de Rennes, France.

I. Good, 1953. "The Population Frequencies of Species and the Estimation of Population Parameters", *Biometrika* 40, 237-264.

R. Kaplan, and J. Bresnan, 1982. "Lexical-Functional Grammar: A Formal System for Grammatical Representation", in J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, The MIT Press, Cambridge, Mass.

R. Kaplan, 1989. "The Formal Architecture of Lexical-Functional Grammar", *Journal of Information Science and Engineering*, vol. 5, 305-322.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz and B. Schasberger, 1994. "The Penn Treebank: Annotating Predicate Argument Structure". In: *ARPA Human Language Technology Workshop*, 110-115.

J. Maxwell and R. Kaplan, 1991. "A Method for Disjunctive Constraint Satisfaction", in M. Tomita (ed.), *Current Issues in Parsing Technology*, Kluwer Academic Publishers.

H. Ney, S. Martin and F. Wessel, 1997. "Statistical Language Modeling Using Leaving-One-Out", in S. Young & G. Bloothooft (eds.), *Corpus-Based Methods in Language and Speech Processing*, Kluwer Academic Publishers.

S. Riezler, D. Prescher, J. Kuhn and M. Johnson, 2000. "Lexicalized Stochastic Modeling of Constraint-Based Grammars using Log-Linear Measures and EM Training", *Proceedings ACL'2000*, Hong Kong, China.

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