A Data-Oriented Parsing Model for Lexical-Functional Grammar

Rens Bod and Ronald Kaplan

rens@science.uva.nl, kaplan@parc.xerox.com

Submitted for Publication. Comments are wellcome.

Abstract

Data-Oriented Parsing (DOP) models of natural language propose that human language processing works with representations of concrete past language experiences rather than with abstract linguistic rules. These models operate by decomposing the given representations into fragments and recomposing those pieces to analyze new utterances. A probability model is used to select from all possible analyses of an utterance the most likely one. Previous DOP models were based on simple tree representations that neglect grammatical functions and syntactic features (Tree-DOP). In this paper, we present a new DOP model based on the more articulated representations of Lexical-Functional Grammar theory (LFG-DOP). LFG-DOP triggers a new, corpus-based notion of grammaticality, and an interestingly different class of probability models. An empirical evaluation of the model shows that larger as well as richer fragments improve performance. Finally, we go into some of the conceptual implications of our approach.

1 Introduction

During the last few years, a new approach to linguistic analysis has started to emerge. This approach, which has come to be known as Data-Oriented Parsing or DOP[1], embodies the assumption that human language comprehension and production works with representations of concrete past language experiences rather than with abstract linguistic rules. The models that

[1] For examples of work within this framework, see Bender and Riehemann (2000), Bod (1993, 1998), Bod and Kaplan (1997, 1998), Bod et al. (2002), Bonnema et al. (1997), Carroll and Weir (2001), Charniak (1996), Coleman and Pierrehumbert (1997), Collins and Duffy (2001), de Pauw (2000), Goodman (1996), Kaplan (1996), Neumann (1998), Scha (1990), Scha et al. (1999), Sima'an (1999), Way (1999), among others.

instantiate this approach operate by decomposing the given representations into fragments and recomposing those pieces to analyze (infinitely many) new utterances. A probability model is used to select from all possible analyses of an utterance the most likely one.

A DOP model can in principle be defined for every theory of linguistic representation or utterance analysis. Any theory of linguistic representation is usually part of a larger linguistic theory that also provides for rules, derivational mechanisms, and other specifications by which the representations for a particular utterance are determined. The rules of the larger theory are chosen and organized not just in order to make this determination, however. They also carry a burden of scientific explanation. Thus a rule set is evaluated according to how simple the individual rules are, how well they express independent linguistic generalizations, and how freely they interact to produce the set of all possible representations for all possible utterances. In writing a grammar, a linguist is in effect searching for the smallest, nonredundant, orthogonal basis for the whole set of utterance-representation assignments. Grammatical formalisms are intended to aid in this search by limiting the amount of information that a given rule can make reference to.

The empirical challenge for such a pursuit is that some constructions of natural language have dependencies (e.g. special meanings or statistical privileges of occurrence) that cannot be accounted for by the free interaction of smallest independent rules. Idioms and other fixed constructions are the typically recognized examples, but proponents of Construction Grammar observe that constructions with unanalyzable properties are also quite prevalent (Fillmore et al. 1988; Goldberg 1995). The rule formalisms of most linguistic theories embody the smallest/independent bias so strongly that they make it difficult to characterize the special properties of larger units of language. More than that, since larger constructions are usually made up of smaller ones, it is conceptually difficult to decide where to draw the boundaries around a particular set of dependencies.

A DOP model stands in sharp contrast to the usual linguistic approach. A DOP model that incorporates the utterance representations of a given linguistic theory does not incorporate the particular grammatical rules and derivational mechanisms of that theory. And most importantly, it is not at all biased in the direction of smallest/independent specification. A DOP model does not even require the identification of a specific collection of larger constructions; it allows for utterance analyses to be created from corpus structures of arbitrary size and complexity, even from structures that are actually substructures of other ones. A probability model is used to choose from among the collection of different structures of different sizes those that make up the most appropriate analysis of a particular utterance.

Thus, although a DOP model for a given theory of representation will produce utterance analyses that are compatible with that theory, it does not depend on or contribute to the discovery of any set of rules or mechanisms of a conventional "explanatory" theory of language. DOP models based on finite corpora are productive, however, in that they can provide analyses for infinitely many novel utterances. And a DOP model may be regarded as offering an alternative view of what native speakers know when they know a language. Linguistic competence may consist not of a collection of succinctly represented generalizations that characterize a language; rather, competence may be nothing more than probabilistically organized memories of prior linguistic experiences.

In accordance with the general DOP architecture outlined by Bod (1995), the first step in constructing a particular DOP model for a language is to specify settings for the following four parameters:

- a formal definition of a well-formed *representation for utterance analyses,*
- a set of *decomposition operations* that divide a given utterance analysis into a set of *fragments*,
- a set of *composition operations* by which such fragments may be recombined to derive an analysis of a new utterance, and
- a *probability model* that indicates how the probability of a new utterance analysis is computed on the basis of the probabilities of the fragments that combine to make it up.

The second step is to acquire a corpus each of whose utterances is annotated with a well-formed and linguistically most appropriate representation. The third step is to generate the fragments for the given corpus by systematically applying the decomposition operations to each of the corpus representations. A new utterance analysis can then be derived by applying the composition operations to a sequence of the resulting fragments. The probability model is used to rank different analyses of an utterance.

The general DOP architecture thus allows for a wide range of different instantiations. It postulates a probabilistic, corpus-based approach, but it leaves open how the utterance-analyses in the corpus are represented, what the substructures of these utterance-analyses are that play a role in processing new input, and what the details of the probabilistic calculations are. The original DOP model, called Tree-DOP, uses surface phrase-structure trees as its corpus representations (Bod 1992, 1993). Tree-DOP is limited, however, in that its representations do not encode grammatical functions as subject, predicate and object, or agreement features. In this paper, we investigate what is involved in creating a DOP model for linguistically sophisticated

representations as proposed by modern linguistic theories. We have chosen the representations defined by Lexical-Functional Grammar (LFG) theory, since they have been shown to apply to a wide range of languages and linguistic phenomena (cf. Dalrymple et al. 1995; Bresnan 2001). Moreover, the recent availability of LFG-annotated corpora provides an actual test domain for our model.

The rest of this paper is organized as follows. We start with a review of the original Tree-DOP model, and explain the nature of the DOP hypothesis which was put forward in Bod (1998) and which states that parse accuracy increases with increasing fragment size. In section 3, we investigate what is involved in extending Tree-DOP to the representations proposed by LFG theory. The resulting LFG-DOP model triggers a new, corpus-based notion of grammaticality. In section 4, we briefly explain how existing parsing models can be used to test LFG-DOP. In section 5, we report on a number of experiments showing, among other things, that the DOP hypothesis receives further support from LFG-DOP, and that LFG-DOP outperforms Tree-DOP if evaluated on tree structures. Finally, we will go into some of the conceptual implications of our results.

2 Review of Tree-DOP

We begin with a review of the original Tree-DOP model, since this model will be the basis for our DOP model for LFG-representations. The Tree-DOP model was developed by Bod (1992, 1993) although the presentation here is somewhat different from the original. As the name suggests, the linguistic representations used by the Tree-DOP model are standard phrase structure trees that characterize the surface constituent structures of utterances. Tree-DOP admits only trees without nonbranching dominance cycles and thus guarantees that any finite string has only a finite degree of ambiguity.

The decomposition operations of Tree-DOP produce connected subtrees of an utterance representation. Tree-DOP has 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.

For example, suppose we have the tree in figure 1 (we leave out some subcategories to keep the example simple).

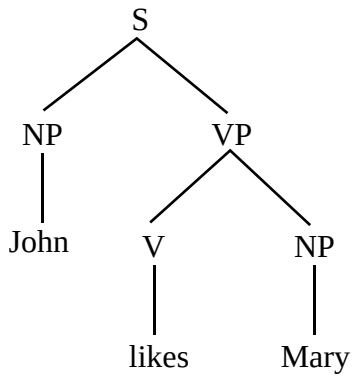

Figure 1. An example tree for *John likes Mary*

Then the result of applying the *Root* operation to the VP-labeled node above is the subtree

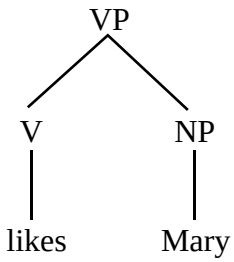

Figure 2. A fragment of the tree in figure 1

Applying the *Frontier* operation to the node sets {NP} and {V, NP} gives the respective fragments

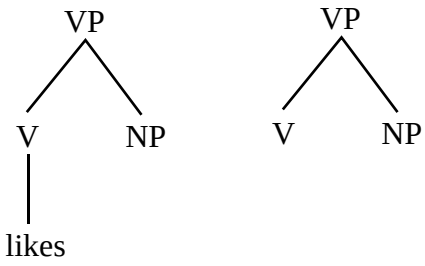

Figure 3. Two other fragments

Note that the decomposition operations exclude fragments such as

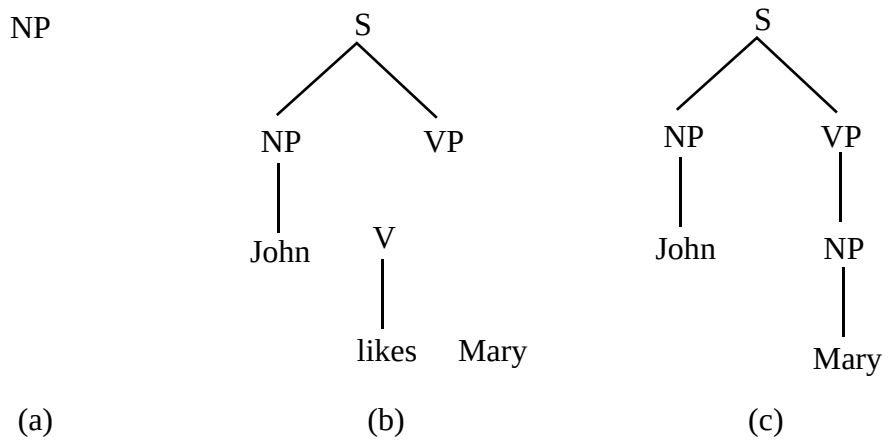

Figure 4. Example of non-valid fragments

Subtree (a) is not produced because *Frontier* cannot choose a subtree's root node, and the disconnected structure in (b) is not produced because *Frontier* erases complete subtrees. Finally, (c) is excluded because *Frontier* erases *all* subtrees dominated by a chosen node. The fact that *Frontier* is defined to delete all daughter subtrees has the effect of preserving the integrity of subcategorization dependencies that are typically encoded as sister relations in phrase structure representations.

Tree-DOP specifies only one composition operation, a node-substitution operation that replaces the left-most nonterminal frontier node in a subtree with a fragment whose root category matches the category of the frontier node. The composition operator is notated by ∘ and its effect is illustrated in figure 5:

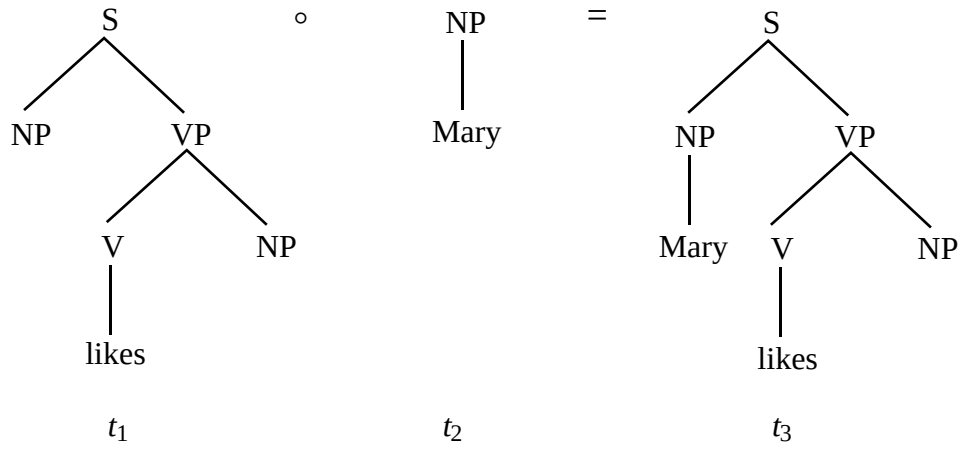

Figure 5. Illustration of the composition operation

The resulting tree t_3 can be composed with another NP fragment t_4 to derive an analysis t_5 for the sentence *Mary likes John*:

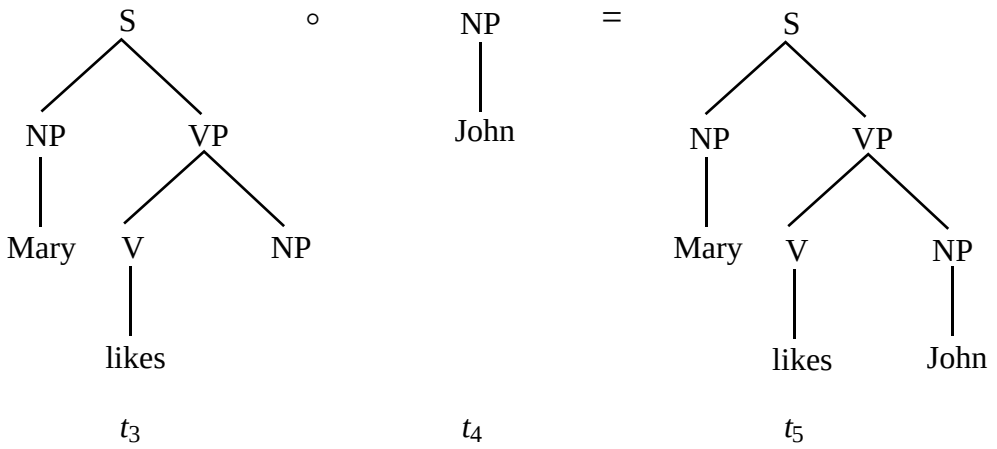

Figure 6. Deriving an analyis for *Mary likes John*

Thus the sequence of fragments $<t_1, t_2, t_4>$ is a derivation for the analysis t_5 under the composition operator ∘; this fact can be written as the expression $t_1 \circ t_2 \circ t_4 = t_5$ with the convention that ∘ is left-associative. Notice that the representation t_5 can also be derived from other sequences of fragments that the decomposition operations produce from the tree in figure 1. For example

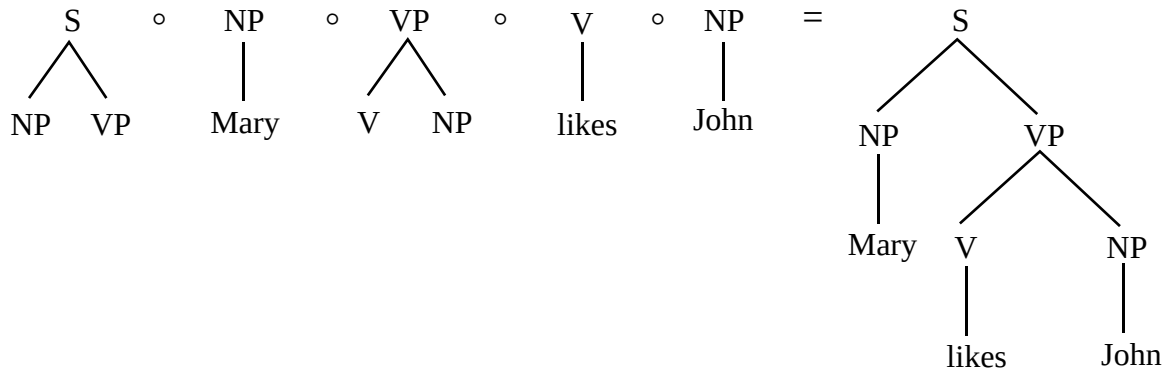

Figure 7. Another derivation for the same analysis for *Mary likes John*

This particular Tree-DOP derivation corresponds to a context-free derivation in that each of its fragments is equivalent to a simple context-free rule. Thus there is a considerable amount of spurious ambiguity in Tree-DOP, in that there are different derivations for a given tree -- not to be confused with structural ambiguity when there are different trees for a given sentence.

The fact that there are typically many different derivations for a given representation R is the major point of departure for the Tree-DOP probability model. If each derivation D has a probability P(D), then the probability of deriving R is the sum of the individual derivation probabilities:

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

A Tree-DOP derivation $D = \langle t_1, t_2 \ldots t_k \rangle$ is produced by a stochastic branching process. It starts by randomly choosing a fragment t_1 labeled with the initial category (e.g. S). At each subsequent step, a next fragment is chosen at random from among the set of competitors for composition into the current subtree. The chosen fragment is composed with the current subtree to produce a new one. The process stops when a tree results with no nonterminal leaves. Let CP(t | CS) denote the probability of choosing a tree t from a competition set CS containing t. Then the probability of a derivation is the product of the chosen fragments:

$$(2) \qquad P(\langle t_1, t_2 \ldots t_k \rangle) = \prod_i CP(t_i \mid CS_i)$$

where the competition probability CP(t | CS) is given by

$$(3) \qquad CP(t \mid CS) = \frac{P(t)}{\sum_{t' \in CS} P(t')}$$

Here, P(t) is the fragment probability for t in a given corpus. Let $T_{i-1} = t_1 \circ t_2 \circ \ldots \circ t_{i-1}$ be the subanalysis just before the i^{th} step of the process, let LNC(T_{i-1}) denote the category of the leftmost nonterminal frontier node of T_{i-1}, and let r(t) denote the root category of a fragment t. Then the competition set at the i^{th} step is

$$(4) \qquad CS_i = \{\, t : r(t) = LNC(T_{i-1}) \,\}$$

That is, the competition sets for Tree-DOP are determined by the category of the leftmost nonterminal of the current subanalysis.[2] We observe that at every step in a well-formed

[2] This is not the only possible definition of competition set. As Manning and Carpenter (1997) have shown, the competition sets can be made dependent on the composition operation. Their left-corner language

derivation it is always the case that $LNC(T_{i-1}) = r(t_i)$. This means that the competition set for any fragment t depends only on its root node category $r(t)$ and is thus independent of the derivation it appears in. For Tree-DOP, then, the competition probabilities are simplified to the formula

$$(5) \qquad CP(t \mid CS) = CP(t) = \frac{P(t)}{\sum_{t' : r(t') = r(t)} P(t')}$$

where the fragment probability $P(t)$ is directly estimated by the relative frequency of t in the corpus.

The expressions (1)-(5) define the probability of producing a particular representation in terms of fragment probabilities. We are often interested in the probability distribution for the representations that are assigned to a particular word string W. This distribution is determined by a process that samples just from the subset of representations whose yield is W. The probability of a representation R given that it yields W is defined by the conditional probability

$$(6) \qquad P(R \mid R \text{ yields } W) = \frac{P(R)}{\sum_{R' \text{ yields } W} P(R')}$$

During the past few years, the Tree-DOP model has been extensively evaluated in the context of natural language disambiguation, using standard domains such as the ATIS corpus (e.g. Bod 1993, 1998; Goodman 1998, 2001; Sima'an 1995) and the Wall Street Journal corpus (e.g. Sima'an 2000; Bod 2001). Natural language disambiguation is a hot topic in the field of natural language processing and all state-of-the-art models use nowadays a probabilistic approach to predict the best parse of a sentence (see Manning & Schütze 1999 for an overview). The evaluation method employed by these models is the so-called *blind testing* method (Black et al. 1991). This method randomly divides a corpus of manually disambiguated sentences into a *training set* and a *test set* (usually a 90%/10% division). The analyses from the training set are used to "train" the model, while the sentences of the test set are used as input when the model is tested. The degree to which the most probable analyses generated by the model match with the test set analyses is a measure for the *parse accuracy* of the model.

By systematically testing the effect of various contraints on the fragments that can be derived from the training set, Bod (1993, 1998, 1999, 2001) has observed an interesting

---

model would also apply to Tree-DOP, yielding a different definition for the competition sets. But the properties of such Tree-DOP models have not been investigated.

empirical property which is known as the *DOP hypothesis*. This hypothesis states that *the parse accuracy increases with increasing fragment size*. In other words, any restriction on the fragments from the training set decreases the parse accuracy on the test set. The DOP hypothesis is now widely accepted and has been corroborated by many others, including Sima'an (1995, 1999, 2000), Sekine and Grishman (1995), Bonnema et al. (1997), Poutsma (2000), de Pauw (2000), Chappelier and Rajman (2001), and Collins and Duffy (2001). Unfortunately, the hypothesis is rather self-evident for Tree-DOP. This is because the corpus representations used by Tree-DOP do not encode dependencies between subject, predicate and object, and therefore large tree fragments are usually needed to capture these dependencies.[3] All modern linguistic theories propose more articulated representations in order to characterize such grammatical dependencies. This raises the question whether the DOP hypothesis can also be corroborated for sophisticated linguistic representations. It may very well be the case that for such representations maximal parse accuracy is already achieved by a more restricted set of fragments, or even by the minimal set of smallest fragments (which would correspond to the basic grammar rules of the underlying linguistic theory). In this paper we will develop and test a DOP model for the representations proposed by Lexical-Functional Grammar (LFG) theory. One of the reasons for using LFG representations is the availability of LFG-annotated corpora, thus providing an actual test domain for the DOP hypothesis. Moreover, LFG representations have been shown to apply to a wide range of languages and linguistic phenomena (cf. Dalrymple et al. 1995; Bresnan 2001).

3 A DOP model based on Lexical-Functional representations: LFG-DOP

We now investigate what is involved in extending the Tree-DOP model to the representations of Lexical-Functional Grammar theory (Kaplan and Bresnan 1982). We thus define new settings for the four DOP parameters given in section 1.

---

[3] Some parsing models try to capture these dependencies by associating each constituent label in the surface tree with its headword (e.g. Collins 1999; Charniak 2000). As a consequence, however, these models cannot capture dependencies that involve *non*-headwords of constituents, such as between an adjective and a preposition in *an **obvious** rule **to** everybody*. Tree-DOP, on the other hand, can easily capture this dependency by a subtree which has *obvious* and *to* as its only lexical items. Bod (1998, 2001) gives for a more extensive criticism to so-called "head-lexicalized" parsing models.

Representations

The definition of a well-formed representation for utterance-analyses is 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.)

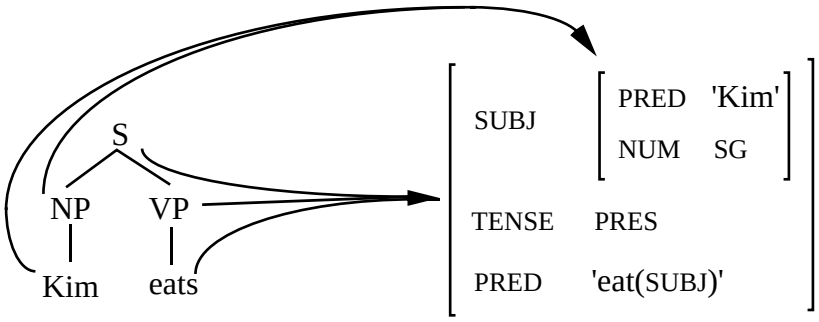

Figure 8. An LFG representation for *Kim eats*

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 8 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 8 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). Note that Completeness and Coherence are the means by which LFG enforces the subcategorization requirements of particular predicates.

Decomposition operations and Fragments

Many different DOP models are compatible with the system of LFG representations (cf. Kaplan 1996). In this paper we outline a basic LFG-DOP model which extends the operations of Tree-DOP to take correspondences and f-structure features into account. The decomposition operations for this model will produce fragments of the composite LFG representations. These will consist of connected subtrees whose nodes are in ϕ-correspondence with sub-units of f-structures. We extend the *Root* and *Frontier* decomposition operations of Tree-DOP so that they also apply to the nodes of the c-structure 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 8, then the f-structure corresponding to the S node is erased, giving figure 9 as a possible fragment:

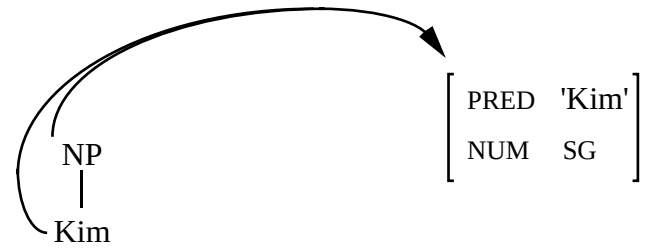

Figure 9. A fragment obtained by the *Root* operation

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:

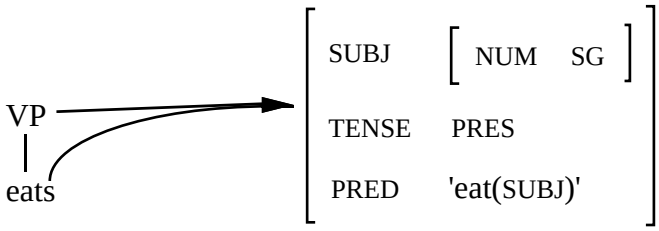

Figure 10. Another *Root*-generated fragment

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 8 is selected as a frontier node, *Frontier* erases the predicate "eat(SUBJ)" from the fragment:

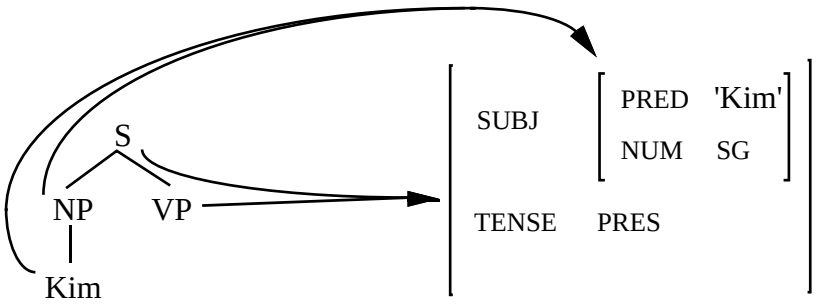

Figure 11. A fragment obtained by the *Frontier* operation

Note that the *Root* and *Frontier* operations retain the subject's NUM feature in the VP-rooted fragment of figure 10, 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 11 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 11 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 11 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 12.

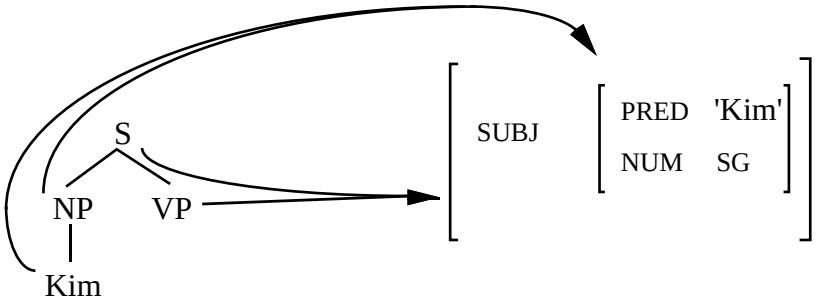

Figure 12. A fragment obtained by the *Discard* operation

The fragment in figure 12 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 12 results from applying *Discard* to the TENSE feature in figure 11. *Discard* also produces fragments such as figure 13, where the subject's number in figure 10 has been deleted:

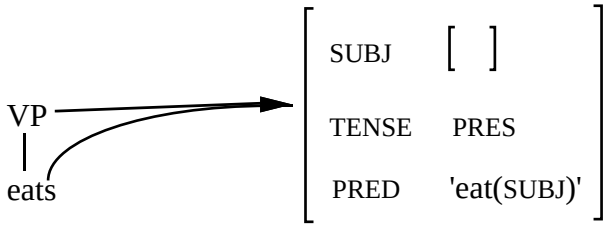

Figure 13. Another fragment obtained by the *Discard* operation

Again, since we have no language-specific knowledge apart from the corpus, we have no basis for ruling out fragments like figure 13. 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.

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 8 for the sentence *Kim eats* and the representation in figure 14 for the sentence *John fell*.

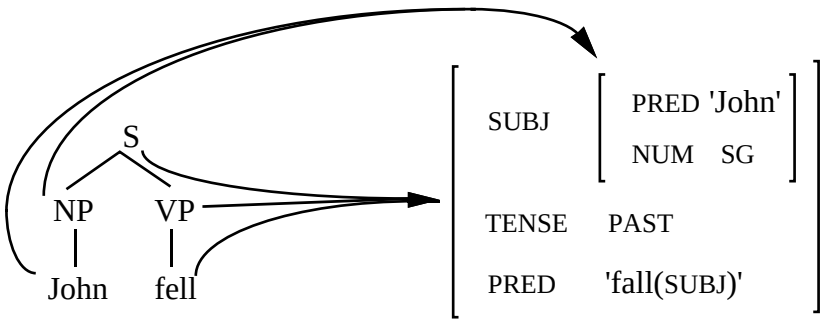

Figure 14. A representation for *John fell*

Figure 15 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*.

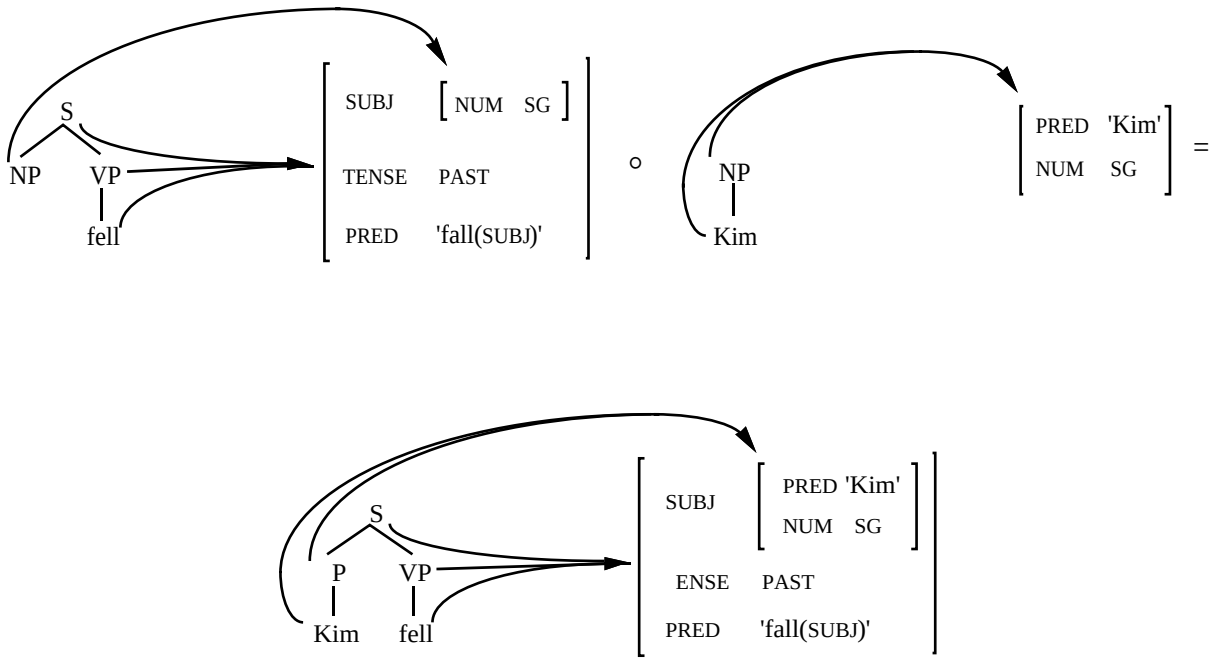

Figure 15. Illustration of the LFG-DOP composition operation

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 16 for the sentence *People ate*.

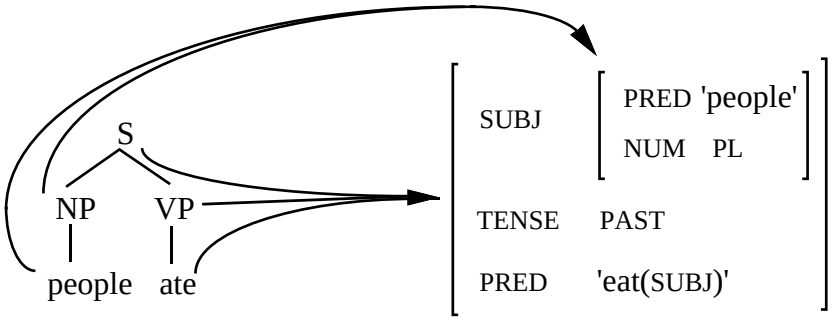

Figure 16. A representation for *People ate*

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*):

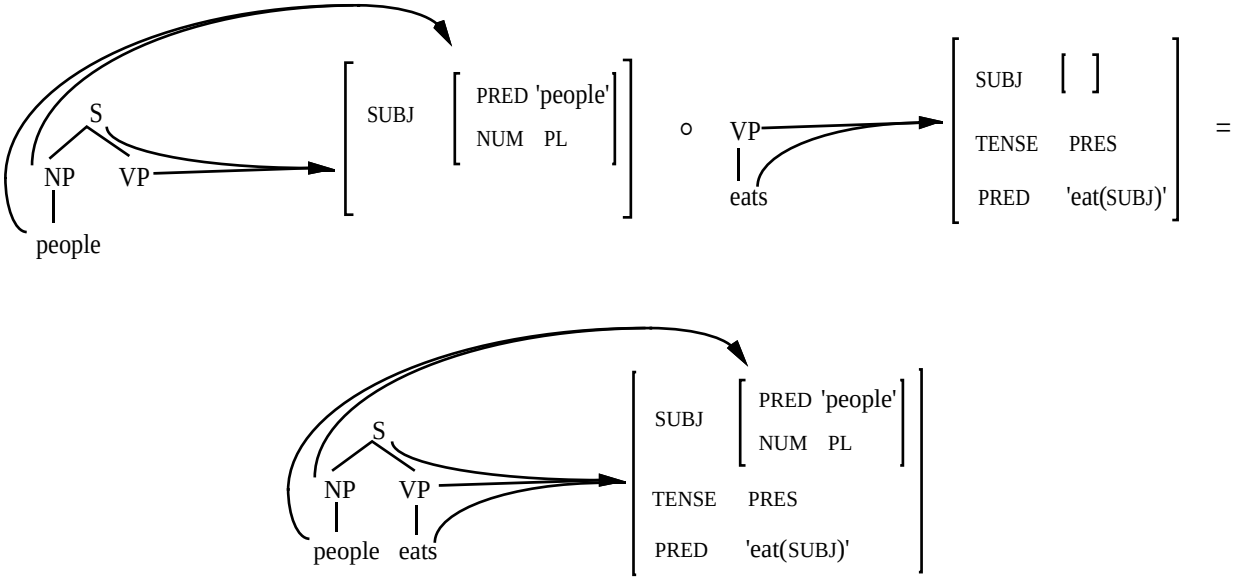

Figure 17. A valid representation for an intuitively ungrammatical sentence

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).

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:

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

An LFG-DOP derivation is also 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. As with Tree-DOP, 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

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

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

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

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

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

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:

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

The formulas (7) through (11) 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

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

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

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

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 (10) and (11) 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

(14) $CS_i = \{ f : r(f) = LNC(F_{i-1})$ and f is unifiable with the 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:

$$(15) \quad CP(f_i \mid CS_i) = \frac{P(f_i)}{\sum_{f:\, r(f)=r(f_i) \text{ and } f \text{ is unifiable with } F_{i-1}} P(f)}$$

On this model we again rely on sampling and the conditional probabilities (10) and (11) 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 (15). 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 5 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), the probability of a fragment was simply defined 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 Bod and Kaplan (1998) showed 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), they did not perform any empirical evaluation. In this paper, we will assess their simple RF estimator in section 5.

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:

$$(16) \quad P(f \mid f \text{ is generated by } \textit{Root/Frontier}) = (1 - n_1/N) \frac{|f|}{\sum_{f'\text{: } f' \text{ is generated by } \textit{Root/Frontier}} |f'|}$$

$$(17) \quad P(f \mid f \text{ is generated by } \textit{Discard}) = (n_1/N) \frac{|f|}{\sum_{f'\text{: } f' \text{ is generated by } \textit{Discard}} |f'|}$$

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 maximum likelihood or log-linear models to LFG-DOP (Berger et al. 1996; Riezler et al. 2000) will be explored in the future.[4]

4 Parsing with LFG-DOP: selecting the most probable analysis

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 14 is indexed as

(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 applying the Tree-DOP decomposition operations described in section 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.

---

[4] 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.

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. This technique is known as "Monte Carlo disambiguation" and has been extensively described in the literature (e.g. Bod 1995, 1998; Chappelier & Rajman 1998, 2000; Goodman 1998; Scha et al. 1999). 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 section 3, 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. We set this error probability to 0.05 in our experiments. An upper bound for this error probability is given by $\sum_{i \neq 0} (1 - (\sqrt{p_0} - \sqrt{p_i})^2)^N$, where the different values of i are indices corresponding to the different analyses, 0 is the index of the most probable analysis, p_i is the probability of analysis i; and N is the number of derivations that was sampled (see Bod 1998: 45-50). This upper bound on the probability of error becomes small if we increase N, but if there is an i with p_i close to p_0, we must make N very large to achieve this effect. Moreover, 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 of N = 10,000 derivations.

5 Empirical Evaluation of LFG-DOP

For our evaluation of LFG-DOP under model M3 we used the (only) two LFG-annotated corpora that are currently available: the Verbmobil corpus, which contains appointment planning dialogues, and the Homecentre corpus, which contains Xerox printer documentation.

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 (i.e. disambiguated) 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 and used paired *t*-testing for evaluating statistical significance between different results.

There is an important question as to what kind of evaluation metric is most appropriate to compare the parses proposed by LFG-DOP with the correct parses in the test set. The most straightforward metric is the so-called *exact match* metric, which is the percentage of proposed parses that exactly match the correct parses. However, it is often the case that a parse is nearly correct except for just one or a few consituents. In such cases it may be interesting to also use a weaker evaluation scheme which evaluates a parse on a constituent basis rather than on a full match basis. Such an evaluation scheme is known as the PARSEVAL scheme, which is based on the notions of *precision* and *recall* and which is widely used in phrase-structure parsing (see Black et al. 1991).[5] PARSEVAL compares a proposed parse *P* with the corresponding correct test set parse *T* as follows:

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

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

[5] One of the reasons for this popularity is the difficulty of achieving an exact match. A typical sentence from the Wall Street Journal (WSJ) corpus in the Penn Treebank (Marcus et al. 1994) has thousands of different possible tree structures, which makes it particularly hard to select the tree which is correct for all its constituents (see Manning and Schütze 1999). State-of-the-art parsing systems achieve an exact match score between 30% and 40% on the WSJ, while their precision and recall scores are much higher and lie roughly between 87% and 91% (see Bod 2001).

According to the original PARSEVAL scheme for phrase-structure trees, a constituent in *P* is "correct" if there exists a constituent in *T* of the same label that spans the same words. In order to apply the PARSEVAL scheme to LFG representations we have extended the 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 the same words *and that ϕ-corresponds to the same f-structure unit*. Note, however, that this evaluation scheme is still quite strong for LFG analyses: while it allows for the weaker PARSEVAL measure on the c-structure trees, it still demands an exact match for the ϕ-corresponding f-structure units. A major aspect of comprehension is the recovery of the predicate-argument relations expressed in a sentence, and these are encoded as the semantic forms in LFG f-structures. Thus, as an indicator of the accuracy of predicate-argument recovery independent of the superficial details of the c-structure (like the PARSEVAL measures) and of other purely syntactic features in the f-structure, we also defined measures of *semantic precision* and *semantic recall*. Under this semantic scheme, a constituent in *P* is correct if there exists a constituent in *T* of the same label that spans the same words *and whose ϕ-corresponding f-structure unit has the same semantic forms*.

5.1 Comparing the two fragment estimators

In our first experiment we were interested in comparing the performance of the simple relative frequency (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 this 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. We will limit ourselves to evaluating the exact match metric and the precision and recall metrics; the semantic metrics will be evaluated in section 5.2.

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 the Verbmobil corpus 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 the Homecentre corpus 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 preferably be viewed as "previously unobserved fragments" whose probability can 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 ($p < 0.05$). 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 that 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 ($p < 0.02$)). Thus for Verbmobil sentences, the use of simple RF as a fragment estimator does not perform significantly better than picking a parse by chance.

5.2 Testing the DOP hypothesis: comparing different fragment sizes

Next, we were interested in testing the DOP hypothesis (see section 2) 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 component. 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, where we also evaluated on semantic precision (SemPrecision) and semantic recall (SemRecall).

Size	Exact Match	Precision	Recall	SemPrecision	SemRecall
1	30.6%	74.2%	72.2%	83.3%	80.8%
≤2	34.1%	76.2%	74.5%	86.9%	82.7%
≤3	35.6%	76.8%	75.9%	87.8%	85.3%
≤4	35.9%	77.5%	76.4%	88.1%	86.7%

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

Size	Exact Match	Precision	Recall	SemPrecision	SemRecall
1	31.3%	75.0%	71.5%	84.8%	80.7%
≤2	36.3%	77.1%	74.7%	87.4%	84.5%
≤3	37.8%	77.8%	76.1%	89.0%	86.1%
≤4	38.4%	80.0%	78.6%	90.5%	87.4%

Table 4. Accuracies on the Homecentre for different fragment sizes

The tables show that there is an increase in accuracy for all metrics if larger fragments are included. This result is significant in that it extends the plausibility of the DOP hypothesis to the more sophisticated LFG representations. According to paired *t*-testing, all differences between the minimal and maximal accuracies for each metric are statistically significant (all with a significance level of 0.001 or lower). Note that the semantic precision/recall metrics are consistently higher than the other precision and recall metrics. This result is obvious since the semantic metrics only evaluate on the semantic forms, while the other metrics also take into account the syntactic features in the f-structures.

5.3 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 in table 6 are statistically significant (with a significance level of 0.01 or lower). Although this result may not seem very surprising, it is important because most parsing models are still evaluated on tree structures only (cf. Collins 1999, 2000; Charniak 2000; Manning and Schütze 1999). Since Tree-DOP obtains very competitive accuracy on the standard Wall Street Journal corpus in the Penn Treebank (see Bod 2001), LFG-DOP may further improve the parse accuracy if the functional annotations in the Penn Treebank (Marcus et al. 1994) can be converted into LFG-style functional structures.

6 Conclusion

We have developed a Data-Oriented Parsing model based on the syntactic representations of Lexical-Functional Grammar theory: LFG-DOP. 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, which suggests that generalized fragments should be treated as previously unseen fragments. We have also seen that LFG's functional structures contribute to higher parse accuracy on tree structures, and that the DOP hypothesis, which states that parse accuracy increases with increasing fragment size, can be corroborated for LFG representations. We do not know of any other work that has tested the DOP hypothesis for representations richer than simple tree structures. In Neumann (1998) and Neumann & Flickinger (1999), DOP models are proposed for Tree-Adjoining Grammar and Head-driven Phrase Structure Grammar, but no experiments with different fragment sizes are reported.

We should keep in mind that our experimental results, albeit statistically significant, were obtained on relatively small corpora. One of our future goals is to obtain 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 on the training data. We also intend to find linguistic constraints on the *Discard* operation, a direction which is suggested by Way (1999).

We have proposed a new, corpus-based notion of grammaticality, according to which a sentence is grammatical if it can be generated without generalized fragments. While LFG-DOP takes disambiguation and comprehension as the major behaviors it seeks to account for, it can thus also give an account of grammaticality judgments, which in practice are often taken as the primary empirical constraints on linguistic theories. LFG-DOP also supports an alternative view of what native speakers know when they know a language. Linguistic competence may consist not of a collection of succinctly represented generalizations that characterize a language; rather, competence may be nothing more than probabilistically organized memories of prior linguistic experiences. According to this view, the central concern of linguistics would not be Universal Grammar but defining a Universal Representation for linguistic experiences. The problem of language acquisition would be the problem of acquiring examples of representations of linguistic experiences guided by the Universal Representation formalism. And if there is anything innate in the human language faculty, it would be the Universal Representation for linguistic experiences together with the capacity to take apart and recombine these experiences.

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.

E. Bender and S. Riehemann, 2000. Experience-based HPSG. *Proceedings Berkeley Formal Grammar Conference 2000*, Berkeley, Ca.

M. van den Berg, R. Bod and R. Scha, 1994. A Corpus-Based Approach to Semantic Interpretation. *Proceedings Ninth Amsterdam Colloquium*, Amsterdam, The Netherlands.

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, 1992. A Computational Model of Language Performance: Data-Oriented Parsing. *Proceedings COLING'92*, Nantes, France.

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 1999. Context-Sensitive Spoken Dialogue Processing with the DOP Model. *Natural Language Engineering* 5(4), 309-323.

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, 1997. On Performance Models for Lexical-Functional Analysis. *Computational Psycholinguistics Conference 1997*, Berkeley, Ca.

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

R. Bod, R. Scha and K. Sima'an, 2001. *Data-Oriented Parsing*. CSLI Publications, Stanford. (in press)

R. Bod, J. Hay and S. Jannedy, 2002. *Probability Theory in Linguistics*. The MIT Press, Cambridge. (in press)

R. Bonnema, R. Bod and R. Scha, 1997. A DOP Model for Semantic Interpretation. *Proceedings ACL/EACL-97*, Madrid, Spain.

J. Bresnan, 2001. *Lexical-Functional Syntax*. Blackwell, Oxford.

J. Carroll and D. Weir, 2001. Encoding Frequency Information in Lexicalized Grammars. in Bod et al., 2001.

E. Charniak, 1996. Treebank Grammars. *Proceedings AAAI-1996*, Menlo Park, Ca.

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

J. Chappelier and M. Rajman, 1998. Extraction stochastique d'arbres d'analyse pour le modèle DOP. *Proceedings TALN'98*, Paris, France.

J. Chappelier and M. Rajman, 2000. Monte Carlo Sampling for NP-hard Maximization Problems in the Framework of Weighted Parsing. in *Natural Language Processing -- NLP 2000, Lecture Notes in Artificial Intelligence 1835*, D. Christodoulakis (ed.), 2000, 106-117.

J. Chappelier and M. Rajman, 2001. Parsing DOP with Monte Carlo Techniques. in Bod et al., 2001.

J. Coleman and J. Pierrehumbert, 1997. Stochastic Phonological Grammars and Acceptability. *Proceedings Computational Phonology, Third Meeting of the ACL Special Interest Group in Computational Phonology*, Madrid, Spain.

M. Collins, 1999. *Head-Driven Statistical Models for Natural Language Parsing*. PhD thesis, University of Pennsylvania, PA.

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

M. Collins and N. Duffy, 2001. Convolution Kernels for Natural Language. *Proceedings Neural Information Processing Systems 2001 (NIPS 2001)*, Alberta, Canada.

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.

M. Dalrymple, R. Kaplan, J. Maxwell and A. Zaenen (eds.), 1995. *Formal Issues in Lexical-Functional Grammar*. CSLI Publications, Stanford.

C. Fillmore, P. Kay and M. O'Connor, 1988. Regularity and Idiomaticity in Grammatical Constructions: The Case of *let alone*. *Language* 64: 501-38.

A. Goldberg, 1995. *Constructions*. Chicago: University of Chicago Press.

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

J. Goodman, 1996. Efficient Algorithms for Parsing the DOP Model. *Proceedings Empirical Methods in Natural Language Processing*, Philadelphia, PA.

J. Goodman, 2001. Efficient Parsing of DOP with PCFG-Reductions. in Bod et al., 2001.

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.

R. Kaplan, 1996. A Probabilistic Approach to Lexical-Functional Analysis. *Proceedings of the 1996 LFG Conference and Workshops*, CSLI Publications, Stanford, Ca.

C. Manning and B. Carpenter, 1997. Probabilistic parsing using left corner language models. *Proceedings IWPT'97*, Boston (Mass.).

C. Manning and H. Schütze, 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge.

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.

G. Neumann, 1998. Automatic Extraction of Stochastic Lexicalized Tree Grammars from Treebanks. *Proceedings of the 4th Workshop on Tree-Adjoining Grammars and Related Frameworks*, Philadelphia, PA.

G. Neumann and D. Flickinger, 1999. Learning Stochastic Lexicalized Tree Grammars from HPSG. DFKI Technical Report, Saarbrücken, Germany.

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.

G. de Pauw, 2000. Aspects of Pattern-matching in Data-Oriented Parsing. *Proceedings COLING-2000*, Saarbrücken, Germany.

A. Poutsma, 2000. Data-Oriented Translation. *Proceedings COLING-2000*, Saarbrücken, Germany.

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.

R. Scha, 1990. Taaltheorie en Taaltechnologie; Competence en Performance. In Q. de Kort and G. Leerdam (eds.), *Computertoepassingen in de Neerlandistiek*, Almere: Landelijke Vereniging van Neerlandici (LVVN-jaarboek).

R. Scha, R. Bod and K. Sima'an, 1999. A Memory-Based Model of Syntactic Analysis: Data-Oriented Parsing. *Journal of Experimental and Theoretical Artificial Intelligence* 11 (Special Issue on Memory-Based Language Processing).

S. Sekine and R. Grishman, 1995. A Corpus-based Probabilistic Grammar with Only Two Non-terminals. *Proceedings Fourth International Workshop on Parsing Technologies*, Prague, Czech Republic.

K. Sima'an, 1995. An optimized algorithm for Data Oriented Parsing. in R. Mitkov and N. Nicolov (eds.), *Recent Advances in Natural Language Processing 1995*, volume 136 of *Current Issues in Linguistic Theory*. John Benjamins, Amsterdam.

K. Sima'an, 1997. Explanation-Based Learning of Data-Oriented Parsing. in T. Ellison (ed.) *CoNLL97: Computational Natural Language Learning*, ACL'97, Madrid, Spain.

K. Sima'an, 1999. *Learning Efficient Disambiguation*. PhD thesis, ILLC dissertation series number 1999-02. Utrecht / Amsterdam.

K. Sima'an, 2000. Tree-gram Parsing: Lexical Dependencies and Structural Relations. *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)