Exemplar-based syntax: How to get productivity from examples[1]

RENS BOD

Abstract

Exemplar-based models of language propose that human language production and understanding operate with a store of concrete linguistic experiences rather than with abstract linguistic rules. While exemplar-based models are well acknowledged in areas like phonology and morphology, common wisdom has it that they are intrinsically flawed for syntax where infinite generative capacity is needed. This article shows that this common wisdom is wrong. It starts out by reviewing an exemplar-based syntactic model, known as Data-Oriented Parsing, or DOP, which operates on a corpus of phrase-structure trees. While this model is productive, it is inadequate from the point of grammatical *productivity. We therefore extend it to the more sophisticated linguistic representations proposed by Lexical-Functional Grammar theory, resulting in the model known as LFG-DOP, which does allow for meta-linguistic judgments of acceptability. We show how DOP deals with first language acquisition, suggesting a unified model for language learning and language use, and go into a number of syntactic phenomena that can be explained by DOP but that challenge rule-based models. We argue that if there is anything innate in language cognition it is not Universal Grammar but "Universal Representation".*

1. Many thanks to Susanne Gahl and Alan Yu for organizing the LSA'05 workshop "Exemplar-based models in linguistics" without which this article would not have been written. I am also grateful to Mike Dowman and Tom Wasow, and Susanne, for their excellent comments and suggestions. This is the first article that presents DOP for a general linguistic audience. Many of the ideas were previously published from a computational perspective in Bod (1992, 1998), Bod and Kaplan (1998) and Bod et al. (2003a).

The Linguistic Review 23 (2006), 291–320
DOI 10.1515/TLR.2006.012
0167–6318/06/023-0291

1. Introduction

One of the most important challenges for exemplar-based models of language consists in describing the processes that deal with syntactic structure. While exemplar-based approaches are well acknowledged in areas like phonology and morphology, common wisdom has it that they are intrinsically flawed for syntax where infinite generative capacity is needed. This article shows that this common wisdom is wrong. We will review a class of exemplar-based syntactic models known as data-oriented parsing models, showing that they allow for both productivity and meta-linguistic judgments.

A person who knows a language can understand and produce a virtually endless variety of new and unforeseen utterances. In its effort to understand the nature of this "knowledge of language", linguistic theory has used the artificial languages of logic and mathematics as its paradigm sources of inspiration. Much linguistic research proceeds on the assumption that a language is a well-defined formal code – that to know a language is to know a consistent set of rules, also known as a "competence grammar", which determines unequivocally which word sequences belong to the language, and what their pronunciations, syntactic analyses and semantic interpretations are.

The rules of a competence grammar 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 typically 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 utterances. In writing a grammar, a linguist is in effect searching for the smallest, non-redundant, orthogonal basis for the whole set of utterances.

The empirical challenge for such a pursuit is that many 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 usage-based linguistics and construction grammar observe that constructions and multi-word units are extremely frequent, if not ubiquitous, in natural language (Fillmore et al. 1988; Goldberg 1995; Erman and Warren 2000; Barlow and Kemmer 2000; Bybee to appear).

The rule formalisms of most linguistic theories embody the smallest/independent bias so strongly that they make it difficult to characterize these 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. This has led usage-based linguists to conjecture that every token of language use encountered by a speaker/hearer must be registered in memory, and that language understanding and production operates on a store of "exemplars" (e.g., Bybee to

appear; Verhagen 2005; Goldberg 2006). While this massive linguistic memorization has been controversial for quite some time, there is nowadays a large amount of psycholinguistic evidence indicating that language users store virtually every linguistic token they encounter (see Tomasello 2003; Jurafsky 2003 for overviews). In fact, without massive storage of exemplars, frequencies can never accumulate, and conventional ways of speaking cannot be learned.

To dispel dogmatic slumbers from the start it is good to realize that there is no stable definition of "exemplar". The notion of exemplar has been appealed to in psychological investigations of categorization (Nosofsky 1988), phonology (Pierrehumbert 2001), case-based reasoning (Nickles 2003) and even in philosophy of science (Kuhn 1970; Giere 1988). Exemplars can be categories, linguistic representations, problem-solutions or full-fledged scientific explanations. Despite these differences of what an exemplar is and what it does there is also an important convergence, namely that an exemplar should not be identified with the token it represents. Instead, an exemplar is a categorization, classification or analysis of a token. It is this distinction I will also maintain in my article: while a token is an instance of use, an exemplar is a representation of a token. Thus an exemplar in syntax can be a tree structure of an utterance, a feature structure or whatever syntactic representation one wishes to use to convey the syntactic analysis of a particular utterance. It should be kept in mind that multiple tokens of identical instances are represented by multiple (identical) exemplars, so that frequencies are maintained.

But how can we obtain productivity from exemplars? It should be clear that while we may store every *previous* language token as an exemplar, we cannot store all *possible* exemplars in advance. And neither can we freely combine partial exemplars into new word sequences, as most of such sequences would be unacceptable. We thus rephrase the main challenge for an exemplar-based model of syntax as: how can we obtain *grammatical* productivity from exemplars?

One of the first exemplar-based models that addressed this problem is known as *Data-Oriented Parsing* or *DOP* (Scha 1990; Bod 1992, 1998). This model keeps a store of representations of all previous language experiences. It operates by decomposing the given exemplars into fragments and recomposing those pieces to analyze new utterances. For example, if we choose phrase-structure trees as representations, DOP combines subtrees from the given trees into new trees by means of label substitution subject to a category matching condition.

Although a DOP model uses a "theory of linguistic representation" for its exemplars, it stands in sharp contrast to the majority of current approaches in syntax. A DOP model that incorporates the 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. The frequencies of the various structures and substructures are used to determine the most appropriate analysis of an utterance.

Thus DOP employs a distinctive statistical methodology: since we do not know beforehand which structures may be important, we should not constrain or predefine the productive units but *take all, arbitrarily large fragments of previous utterance-analyses and let the statistics decide* (Bod 1992, 1998). The use of probability theory is a straightforward step as it models the notion of frequency in a mathematically precise way. Results from psycholinguistics support the idea that the frequency of occurrence of a structure is an important factor in language comprehension and production. Jurafsky (2003) reviews evidence showing that in language learning, frequencies determine segmentation and generalization. In syntax, frequencies play a role in the gradience of categories and well-formedness judgments (Manning 2003). And in semantics, frequencies affect disambiguation and interpretation. Gahl and Garnsey (2004) even argue that "knowledge of grammar includes knowledge of probabilities of syntactic structures".

Yet, the frequency of occurrence of a structure is not the only factor in language processing; the *recency* of occurrence of a structure seems to be equally important. Short-term recency effects (priming) are well documented in the psycholinguistic literature (e.g., Bock and Loebell 1990). But there are also long-term effects. How can these two effects be combined? In Bod (1998: 109–111) I argued that recency should be treated not as a separate component but as part of the probability model. This may be supported by Bock (1995) who showed that the recent perception or production of a particular structure increases its probability of being used relative to other structures. This finding indeed suggests that recency may be described by a function which adjusts the frequencies of more recently produced and perceived structures upwards while the frequencies of less recently produced and perceived structures are adjusted downwards, possibly down to zero. Thus although DOP assumes that all linguistic experiences are stored, it does not assume that they are always remembered: structures may be forgotten if they are never invoked again. The various quantitative treatments of recency, and their interaction with discourse structure, go beyond the scope of this paper (see Bod 1998), but it is important to keep in mind that factors such as recency can be modeled by probabilities.

This brings me to the following definition of the general DOP framework. The first step in creating a DOP model for a language consists in specifying settings for the following four parameters:

– a 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 and its meaning is computed on the basis of the frequencies of its fragments.

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. Language *comprehension* corresponds to computing the most probable meaning given an utterance, while language *production* consists of computing the most probable utterance given a meaning. The resulting utterance-analysis is added to the corpus, bringing it into a new "state". Thus the DOP approach is congenial to the usage-based or constructionist approach to language but extends it in an important way by providing a formal model that computes new utterances out of previous utterances.

It should be kept in mind that a corpus of annotated utterances is usually not acquired by means of a (competence) grammar but by human annotators who are only given an annotation guideline with some example-representations. The annotators are asked to construct for each sentence in the corpus an analysis that seemed most appropriate in the context in which the sentence was uttered. One could claim that the human annotators use an internalized grammar to annotate the sentences, but in this article I shall argue that a contrasting position is just as viable: humans understand and produce new sentences by combining partial structures from a set of exemplars, without making recourse to a competence grammar. *The only rules that are needed are the rules for decomposing exemplars into fragments and for recomposing these fragments into representations of new sentences*.

But where does the *initial* set of exemplars come from? For methodological reasons we may separate the problem of acquiring a corpus from the use of an already acquired corpus, but any linguistic theory must have a story about first language acquisition. While the generativist approach assumes that language acquisition is guided by a notion of Universal Grammar, there is an increasing body of evidence, both from psycholinguistics (Abbot-Smith and Tomasello, this issue) and computational linguistics (Klein and Manning 2005), that linguistic structure is learned entirely in a statistical, item-based way.

The key idea in statistical models of language acquisition is that word sequences surrounded by equal or similar contexts are candidates to form a cer-

tain constituent. The probability that a certain substring of a sentence constitutes a certain constituent is computed from the substring's frequency and the frequencies of its contexts (Van Zaanen 2001; Klein and Manning 2005). Van Zaanen (2001) shows that higher-level constituents can be learned in this way, including recursive structures. However, these bootstrapping approaches restrict the lexical relations that are taken into account in learning constituents. For example, the model by Klein and Manning (2005) takes into account only statistics of *contiguous* substrings of sentences while it is well known that many lexical dependencies are non-contiguous or structural (i.e., they can be separated by other words or constituents). The only model which takes into account *all* (contiguous and non-contiguous) substrings of sentences and lets the statistics decide is the model by Bod (2005, 2006). This model initially assigns all possible binary trees to a set of sentences and next uses all subtrees, regardless of size or lexicalization, to compute the most probable parse trees for a set of new sentences (there exists an efficient way to do this – see Goodman 2003). Note that such an approach to language acquisition is just another application of the DOP idea: rather than parsing with already learned (sub)trees, we start out to parse with arbitrarily assigned (sub)trees which are next selected on their usefulness in parsing new input.

However, any approach to language acquisition hinges on a definition of linguistic representation – which in the previous paragraph was assumed to be a simple phrase-structure tree. Without a definition of linguistic representation, there is no learning goal and nothing is learned. In Bod (1998) I therefore argued that if there is anything innate in the human language faculty, it must not be Universal Grammar but *Universal Representation*. I will come back to this in the last section of this article. The main goal of this article is, however, not to describe how DOP deals with first language acquisition, but to show how such an exemplar-based approach allows for grammatical productivity. We should nevertheless keep in mind that these two goals are strongly interrelated: *DOP proposes the same model for language acquisition and language processing.* It models acquisition by assigning arbitrary trees to initial sentences and by letting the frequencies decide which subtrees are "most useful" in understanding new input. It models processing by storing these most useful subtrees (such that they do not have to be learned again) and by recombining them in perceiving and producing new utterances.

In the following section, we will start out by explaining how DOP works with surface constituent trees. Although such a model is productive, it will turn out that it is inadequate from the point of grammatical productivity. In Section 3 we propose a simple extension of this DOP model to the more sophisticated linguistic representations used in Lexical-Functional Grammar theory (Kaplan and Bresnan 1982), resulting in a model known as LFG-DOP, which does allow for meta-linguistic judgments of grammaticality. Finally, in Section 4 we

discuss a number of linguistic phenomena that can be explained by the DOP approach but that challenge rule-based approaches.

2. A first Data-Oriented Parsing model: Tree-DOP

The first and arguably simplest DOP model in the literature is known as Tree-DOP (Bod 1992, 1998). On this model each sentence in the corpus is provided with a constituent tree that describes the syntactic surface structure of that sentence. For the sake of simplicity we illustrate Tree-DOP with a corpus of only two trees given in Figure 1, keeping in mind that currently available corpora typically contain hundreds of thousands of trees (cf. Marcus et al. 1994; Abeillé 2003). We have left out some lexical categories in the trees to keep the example simple.

Given the corpus in Figure 1, a new sentence can be derived by combining subtrees from the trees in the corpus. Tree-DOP employs only one operation for combining subtrees, called "label substitution", indicated as ◦. The substitution-operation identifies the leftmost nonterminal leaf node of one subtree with the root node of a second subtree, i.e., the second subtree is substituted on the leftmost nonterminal leaf node of the first subtree provided their categories match. Starting out with the corpus of Figure 1, for instance, the sentence *She saw the dress with the telescope* may be generated as shown in Figure 2.

The sentence *She saw the dress with the telescope* is ambiguous; by combining other subtrees, a different parse tree may be derived, which is analogous to the first rather than to the second corpus sentence, as shown in Figure 3.

Thus subtrees can be of arbitrary size: they can range from just one categorized word to entire sentence-analyses. The smallest subtrees, such as the

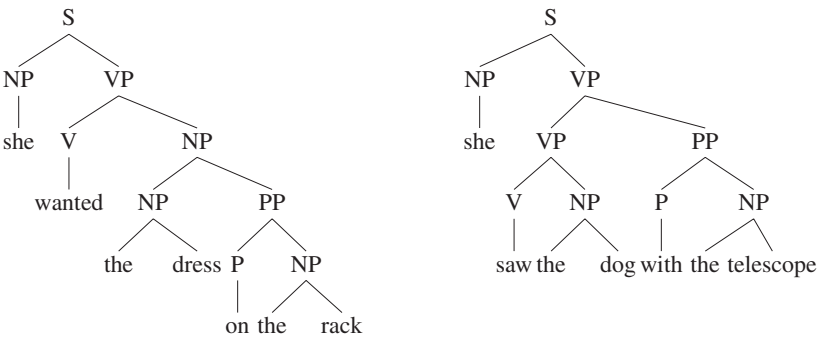

Figure 1. Example corpus of two trees

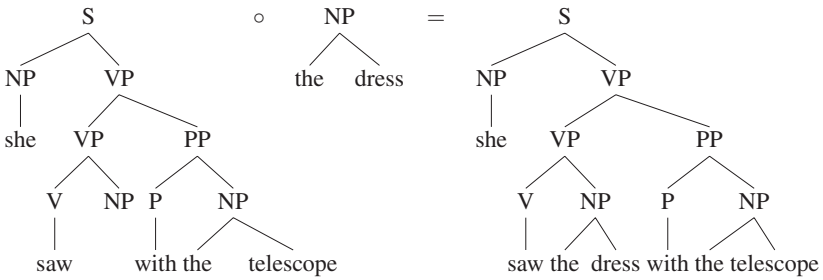

Figure 2. Derivation and parse tree for *She saw the dress with the telescope*

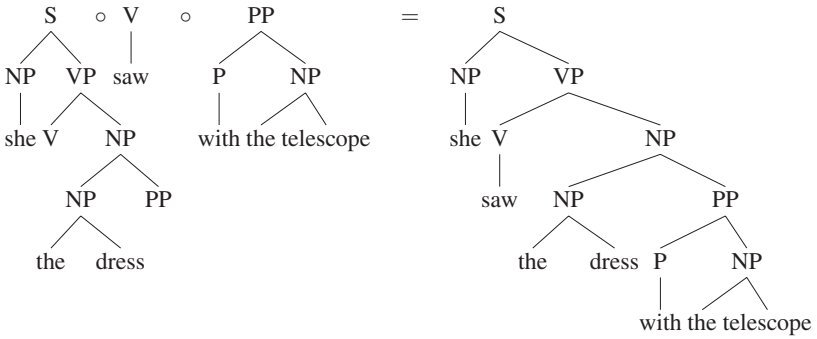

Figure 3. Another derivation and parse tree for *She saw the dress with the telescope*

second subtree in Figure 3, cover only one level of constituent structure and are analogous to "context-free rewrite rules". But in the exemplar-based approach, such a minimal subtree, like other, larger subtrees, are viewed not as rules but as *representations* of a particular word, phrase or construction. Although minimal subtrees may be mathematically isomorphic to a context-free rewrite rule, they do not correspond to the notion of a rule in language as being an "independent linguistic generalization". For example, the subtree $_{\mathrm{V}}$[*saw*] is the annotation of the verb *saw* in the context of the sentence *She saw the dress with the telescope*. There may be other annotations of *saw* but in the example above it corresponds to (a representation of) one instance of language use. It may take some effort to "see" subtrees as representations of concrete language tokens rather than as bunches of rules, possibly involving a Gestalt switch.

Note that Tree-DOP can generate an unlimited number of other sentences on the basis of the corpus in Figure 1, such as *She saw the dress on the rack with*

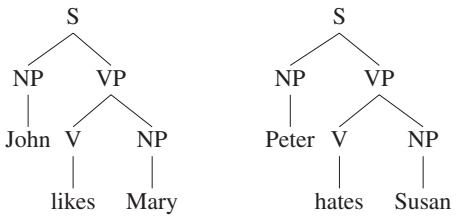

Figure 4. A corpus of two trees

the telescope and *She saw the dress with the dog on the rack with the telescope*, etc. Thus we can indeed get infinite productivity from a finite corpus of exemplars (whether this productivity also corresponds to grammatical productivity will be discussed later). Note also that these sentences are highly ambiguous: several syntactic structures can be assigned to each of them by Tree-DOP. It is the role of Tree-DOP's probability model to select the most probable structure for a certain sentence, or in case of language production, to select the most probable sentence for a certain meaning to be conveyed.

How can we compute the probability of a sentence and its analysis from the frequencies of the corpus subtrees? By having defined a method for combining subtrees from a corpus into new trees, we effectively established a way to view a corpus as a tree generation process. This process becomes a *stochastic* process if we take the frequency distributions of the subtrees into account. For every tree and every sentence we can compute the probability that it is generated by this stochastic process. Before we go into the details of this computation, let us illustrate the stochastic generation process by means of an even simpler corpus than in Figure 1. Suppose that our example corpus consists of two tokens, *John likes Mary* and *Peter hates Susan*, that are represented by their surface constituent trees in Figure 4 (remember that a tree may occur more than once in a corpus; in this simple example corpus each tree occurs only once, while some subtrees occur more than once).

To compute the frequencies of the subtrees in this corpus, we need to define the bag of subtrees that can be extracted from the corpus trees. To this end, we explicitly define Tree-DOP's *decomposition operations* that divide a corpus of trees into its subtrees:

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.

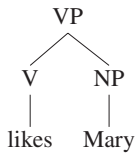

Figure 5. A subtree of the left tree in Figure 4

Figure 6. Two subtrees derived from Figure 5

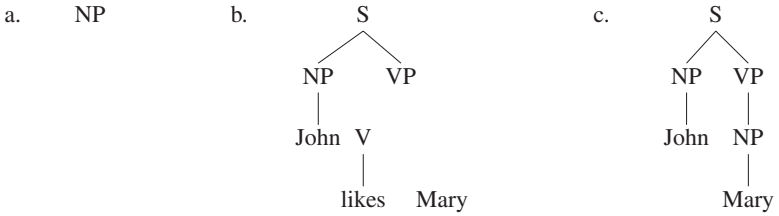

Figure 7. Example of non-valid fragments

As an example let's focus on the left tree in Figure 4 for the sentence *John likes Mary*. The result of applying the *Root* operation to the VP-labeled node in this tree is the subtree in Figure 5.

Applying the *Frontier* operation to, respectively, the node sets {NP} and {V, NP} gives the respective subtrees in Figure 6.

Note that the decomposition operations exclude fragments such as (a)–(c) in Figure 7.

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.

On the basis of these two decomposition operations, the total bag of subtrees that can be extracted from the corpus in Figure 4 is given in Figure 8. Notice that some subtrees occur twice (a subtree may be extracted from different trees

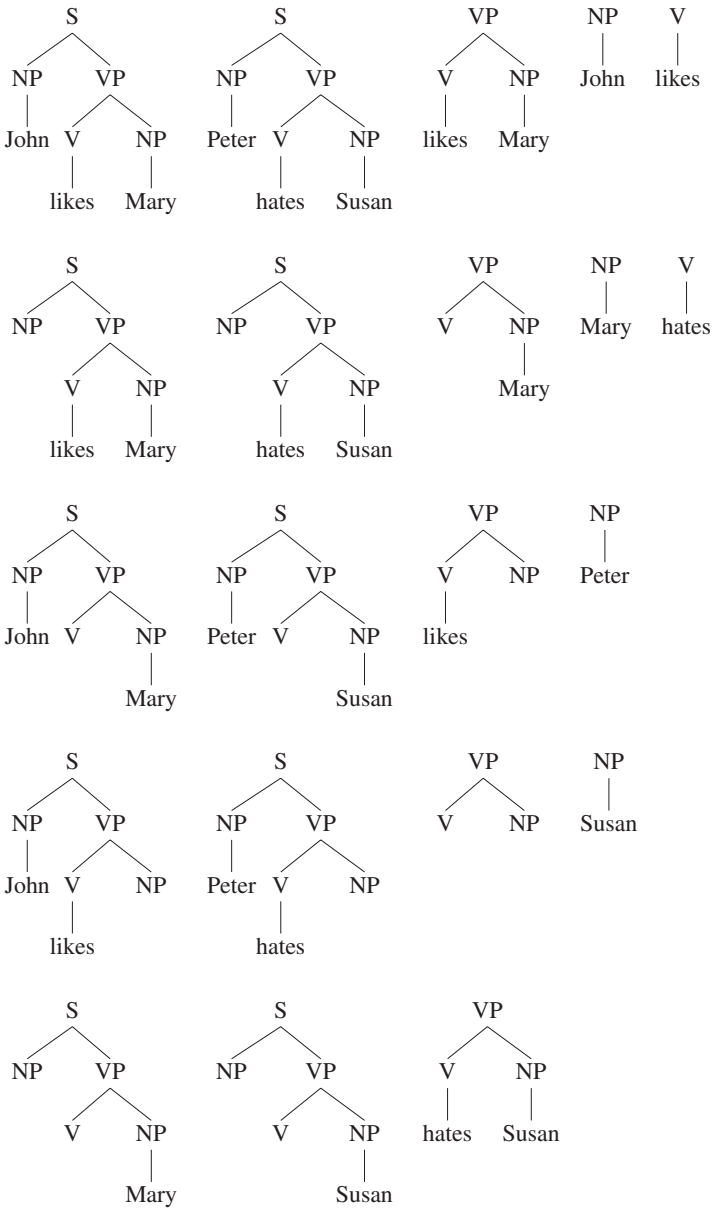

Figure 8. The bag of subtrees derived from the trees in Figure 4

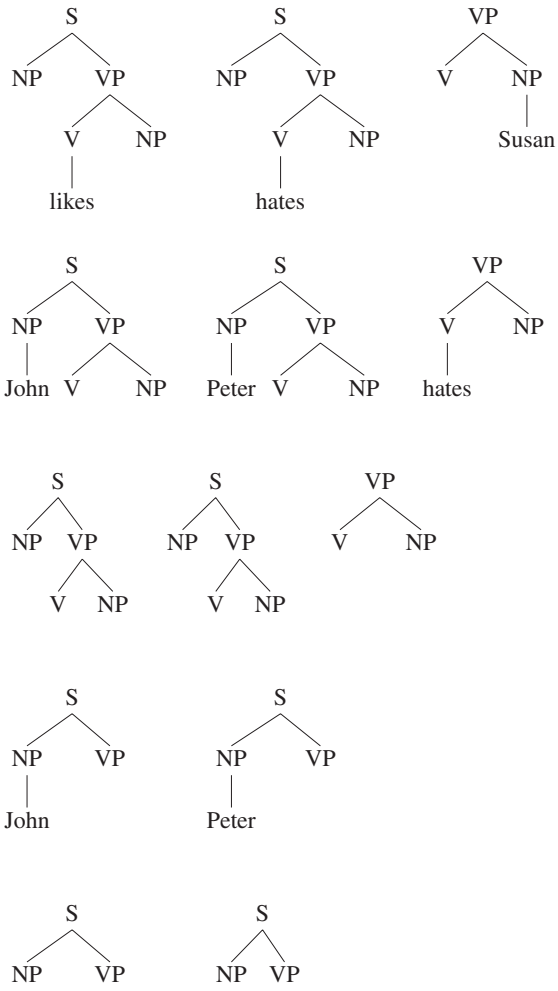

Figure 8 cont. The bag of subtrees derived from the trees in Figure 4

and even several times from a single tree if the same node configuration appears at different positions).

As explained above, by means of the composition operation, new sentence-analyses can be constructed by means of this subtree collection. For instance, an analysis for the sentence *Mary likes Susan* can be generated by combining the three subtrees in Figure 9 from the bag in 8.

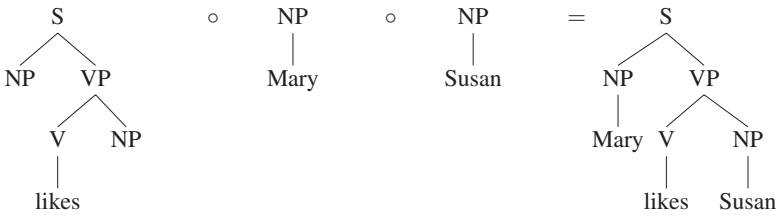

Figure 9. Analyzing *Mary likes Susan* by combining subtrees

For the following it is important to distinguish between a *derivation* and an *analysis* of a sentence. By a derivation of a sentence we mean a sequence of subtrees the first of which is labeled with S and for which the iterative application of the substitution operation produces the particular sentence. By an analysis of a sentence we mean the resulting parse tree of a derivation of the sentence. Then the probability of the derivation in Figure 9 is the *joint* probability of 3 stochastic events (see Bod 2003a for a linguistic introduction to elementary probability theory):

1. selecting the subtree $_{S}$[NP $_{VP}$[$_{V}$[likes] NP]] among the subtrees with root label S,
2. selecting the subtree $_{NP}$[Mary] among the subtrees with root label NP,
3. selecting the subtree $_{NP}$[Susan] among the subtrees with root label NP.

The probability of each event can be computed from the frequencies of the occurrences of the subtrees in the corpus. For instance, the probability of event 1 is computed by dividing the number of occurrences of the subtree $_{S}$[NP $_{VP}$[$_{V}$[likes] NP]] by the total number of occurrences of subtrees with root label S: $\frac{1}{20}$.

In general, let $|\, t \,|$ be the number of times subtree t occurs in the bag and $r(t)$ be the root node category of t, then the probability assigned to t is

$$P(t) = \frac{|t|}{\sum_{t':r(t')=r(t)} |t'|}$$

Since in our stochastic generation process each subtree selection is independent of the previous selections, the probability of a derivation is the product of the probabilities of the subtrees it involves. Thus, the probability of the derivation in Figure 9 is: $\frac{1}{20} \times \frac{1}{4} \times \frac{1}{4} = \frac{1}{320}$.

In general, the probability of a derivation $t_1 \circ \ldots \circ t_n$ is given by

$$P(t_1 \circ \ldots \circ t_n) = \prod_i P(t_i)$$

It should be stressed that the probability of a parse tree is *not* equal to the

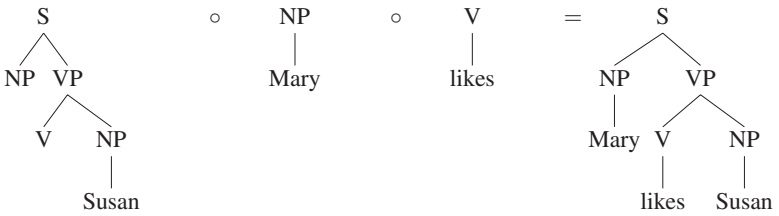

Figure 10. A different derivation, yielding the same parse for *Mary likes Susan*

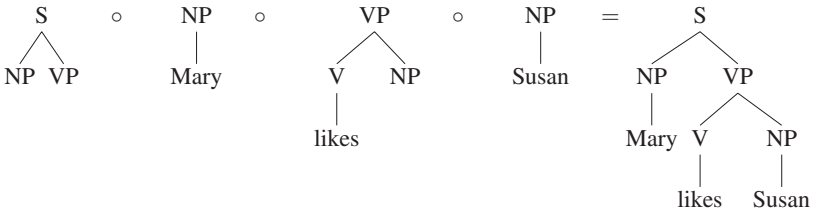

Figure 11. One more derivation yielding the same parse for *Mary likes Susan*

probability of a derivation producing it. There can be many different derivations resulting in the *same* parse tree. This "spurious ambiguity" may seem redundant from a linguistic point of view (and should not be confused with the "structural" ambiguity of a sentence). But from a statistical point of view, all derivations resulting in a certain parse tree contribute to the probability of that tree.

For instance, the parse tree for *Mary likes Susan* derived in Figure 9 may also be derived as in Figure 10 or Figure 11.

Thus, a parse tree can be generated by a large number of different derivations that involve different subtrees from the corpus. Each of these derivations has its own probability of being generated. For example, the following shows the probabilities of the three example derivations given above.

Table 1. Probabilities of the derivations in Figures 9, 10 and 11

$P_{(\text{figure 9})} = \frac{1}{20} \times \frac{1}{4} \times \frac{1}{4} = \frac{1}{320}$
$P_{(\text{figure 10})} = \frac{1}{20} \times \frac{1}{4} \times \frac{1}{2} = \frac{1}{160}$
$P_{(\text{figure 11})} = \frac{2}{20} \times \frac{1}{4} \times \frac{1}{8} \times \frac{1}{4} = \frac{1}{1280}$

The probability of a parse tree is the probability that it is produced by any of its derivations, also called the *disjoint* probability. That is, the probability of a parse tree T is the sum of the probabilities of its distinct derivations D:

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

This step in the calculation does usually not occur when a probabilistically enhanced grammar is used. Such a grammar aims at identifying exactly *one* derivation for each syntactic analysis. This makes computations simpler, but restricts the statistical dependencies beforehand. It should be said though that there is no psycholinguistic evidence that humans take into account multiple derivations in determining the best parse tree for a sentence. Yet, the use of multiple derivations directly follows from the assumption that frequencies of arbitrary lexical relations may be important in syntactic processing. And as shown in the introduction, there is quite some evidence for this assumption.

Analogous to the probability of a parse tree, the probability of an utterance is the probability that it is yielded by any of its parse trees. This means that the probability of a word string W is the sum of the probabilities of its distinct parse trees T:

$$P(W) = \sum_{T \text{ yields } W} P(T)$$

For the task of language comprehension, we are interested in the most probable parse tree given an utterance – or its most probable meaning if we use a corpus in which the trees are enriched with logical forms (and for the task of language production we are interested in the most probable utterance given a certain meaning or logical form). The probability of a parse tree T given that it yields a word string W is computed by dividing the probability of T by the sum of the probabilities of all parses that yield W (i.e., the probability of W):

$$P(T|T \text{ yields } W) = \frac{P(T)}{\sum_{T' \text{ yields} W} P(T')}$$

Since the sentence *Mary likes Susan* is unambiguous with respect to the corpus, the conditional probability of its parse tree is simply 1, by a vacuous application of the formula above. Of course a larger corpus might contain subtrees by which many different representations can be derived for a single sentence, and in that case the above formula for the conditional probability would provide a probabilistic ordering for them. For instance, suppose an example corpus contains the following trees given in Figure 12.

Two different parse trees can then be derived for the sentence *John hates buzzing bees*, given in Figure 13.

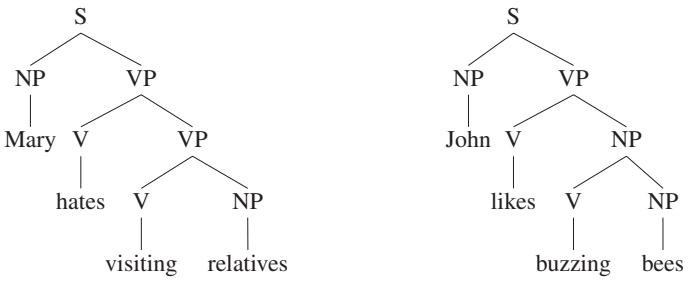

Figure 12. Two corpus trees for *Mary hates visiting relatives* and *John likes buzzing bees*

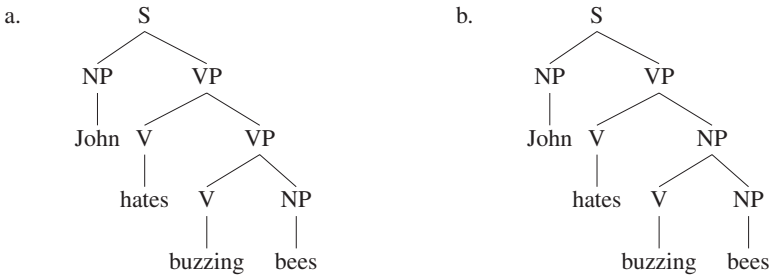

Figure 13. Parse trees for *John hates buzzing bees*

The Tree-DOP model will assign a lower probability to the tree 13 (a) since the sub-analysis 14 (a) of 13 (a) is not a corpus subtree and hence must be assembled from several smaller pieces (leading to a lower probability than when the sub-analysis was a corpus-subtree, since the probabilities of the pieces must be multiplied – remember that probabilities are numbers between 0 and 1). The sub-analysis 14 (b) of 13 (b) can also be assembled from smaller pieces, but it also appears as a corpus fragment. This means that 13 (b) has several more derivations than 13 (a), resulting in a higher total probability (as the probability of a tree is the sum of the probabilities of its derivations).

In general, there tends to be a preference in Tree-DOP for the parse tree that can be generated by the largest number of derivations. Since a parse tree which can (also) be generated by relatively large fragments has more derivations than a parse tree which can only be generated by relatively small fragments, there is also *a preference for the parse tree that can be constructed out of the largest possible corpus fragments*, and thus for the parse tree which is most similar to previously seen utterance-analyses. The same kind of reasoning can be made

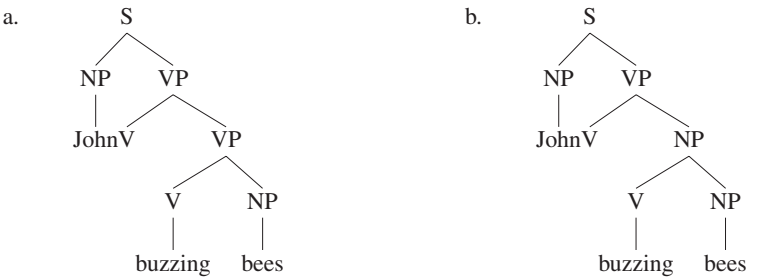

Figure 14. Two sub-analyses

for the probability of an utterance: i.e., there is a preference for the utterance (given a certain meaning or intention) that can be constructed out of the largest possible corpus fragments, thus being most similar to previously seen utterances. This will be particularly important for the reuse of constructions and prefabricated word combinations (see Section 4).

The notion of probability may be viewed as a measure for the *average similarity* between a sentence and the exemplars in the corpus: it correlates with the *number* of corpus trees that share fragments with the sentence, and also with the *size* of these shared fragments. Tree-DOP is thus congenial to analogical approaches to language that also interpret new input analogous to previous linguistic data, such as Skousen (1992) and Daelemans and van den Bosch (2005). DOP differs from analogical approaches in its use of recursive structures. Although analogical approaches have been successful in phonology, morphology and "shallow" parsing, they are intrinsically constrained by their limited generative capacity.

The probability model for Tree-DOP given above is just one of the many possible ways frequencies of previously perceived structures can be taken into account. Other probability models have extended Tree-DOP to recency, semantics and discourse context (Bod 1998, 1999), or use more sophisticated estimation techniques (Zollmann and Sima'an 2005) or even use different definitions of the best parse tree (Bod 2002; Bod et al. 2003). However, the simple probability model described in this section is still the most successful one if tested on benchmarks like the Wall Street Journal treebank (see Bod 2003b for the computational details).

3. Towards a linguistically adequate DOP model: LFG-DOP

While we have shown how to obtain syntactic productivity from exemplars, resulting in a probability model that computes the most similar utterance(-

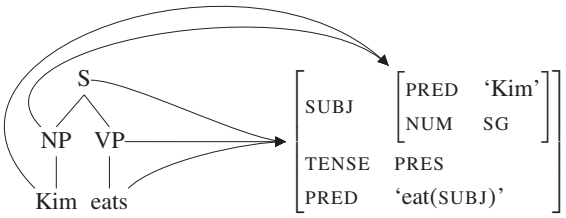

Figure 15. An LFG representation for *Kim eats*

analysis) based on previous utterance-representations, we have not yet shown that we can also get *grammatical* productivity from a corpus of exemplars. In fact, the Tree-DOP model in Section 2 allows for many sentences that would be considered unacceptable to natural language users. This is because our annotations contain only simple syntactic categories with no grammatical functions and no agreement features. As a consequence, Tree-DOP severely overgenerates, producing ill-formed sentences such as *Relatives likes Susan* and *She visiting John*. While this particular overgeneration might be overcome by using more fine-grained subcategorizations, it is well known that the number of categories explode if no agreement features are used (cf. Gazdar et al. 1985).

All modern linguistic theories propose more articulated representations in order to characterize such grammatical relations (cf. Bresnan 2000; Sag et al. 2003). To give an example of how we can construct a linguistically adequate DOP model that not only obtains productivity but also *grammatical* productivity, we will show how the Tree-DOP model can be extended to the representations proposed by a modern linguistic theory, i.e., Lexical-Functional Grammar or LFG (Kaplan and Bresnan 1982). The resulting DOP model is known as LFG-DOP (Bod and Kaplan 1998).

3.1. Representations in LFG-DOP

The notion of representation we will use is directly taken from LFG theory (see Kaplan 1989), 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. Figure 15 shows a representation for the utterance *Kim eats*. (We leave out some features to keep the example simple.)

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 15 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 if and only if 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 15 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. 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.

3.2. Decomposition operations and fragments

We can create a DOP model for LFG-analyses by extending 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 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

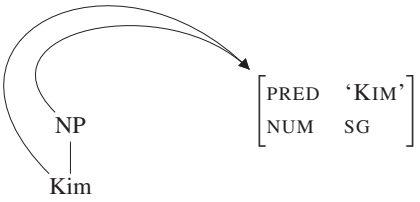

Figure 16. A fragment obtained by the *Root* operation

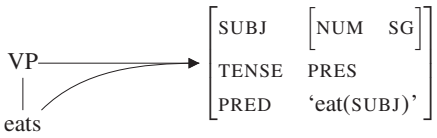

Figure 17. Another *Root*-generated fragment

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 15, then the f-structure corresponding to the S node is erased, giving Figure 16 as a possible fragment:

In addition the *Root* operation deletes from the remaining f-structure all semantic forms 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 17).

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 15 is selected as a frontier node, *Frontier* erases the predicate "eat(SUBJ)" from the fragment (Figure 18).

Note that the *Root* and *Frontier* operations retain the subject's NUM feature in the VP-rooted fragment of Figure 17, 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.

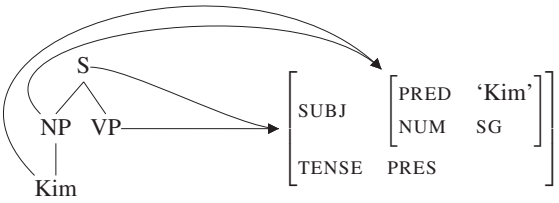

Figure 18. A fragment obtained by the *Frontier* operation

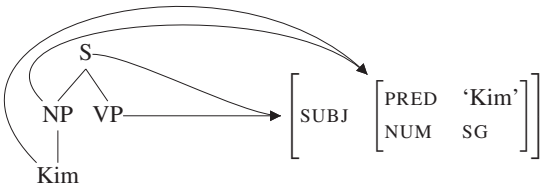

Figure 19. A fragment obtained by the *Discard* operation

On the other hand, the fragment in Figure 18 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 18 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 18 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 19. The fragment in Figure 19 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 19 results from applying *Discard* to the TENSE feature in Figure 18. *Discard* also produces

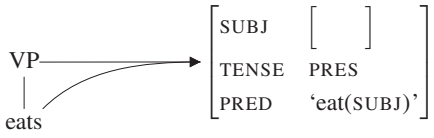

Figure 20. Another fragment obtained by the *Discard* operation

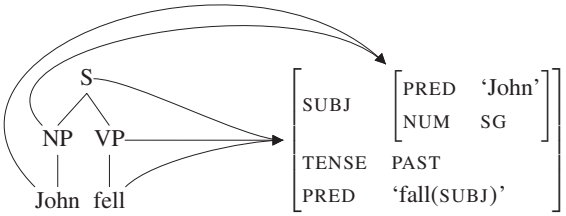

Figure 21. A representation for *John fell*

fragments such as Figure 20, where the subject's number in Figure 17 has been deleted.

Again, since we have no language-specific knowledge apart from the corpus, we have no basis for ruling out fragments like Figure 20. Indeed, it is quite intuitive to omit the subject's number in fragments derived from sentences with past-tense verbs or modals.

3.3. The composition operation

In LFG-DOP the operation for combining fragments, again indicated by $\circ$, is carried out in two steps. First the c-structures are combined by leftmost label 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. For this purpose we assume a corpus containing the representation in Figure 15 for the sentence *Kim eats* and the representation in Figure 21 for the sentence *John fell*.

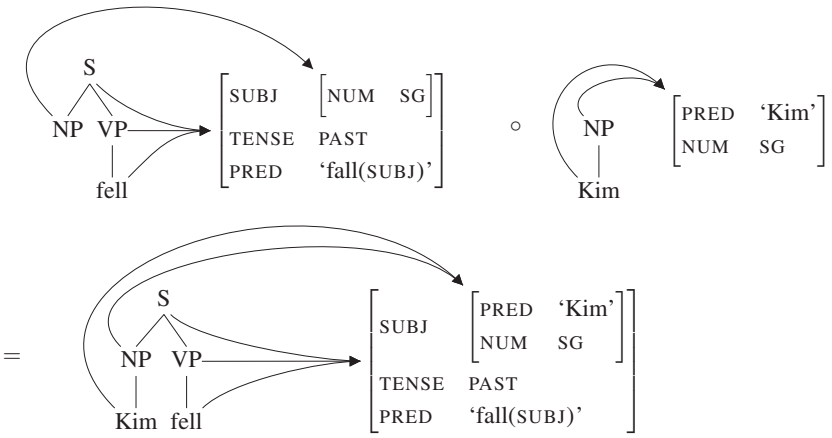

Figure 22. Illustration of the LFG-DOP composition operation

Figure 22 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 in a representation for the new sentence *Kim fell*.

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 23 for the sentence *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*):

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.)

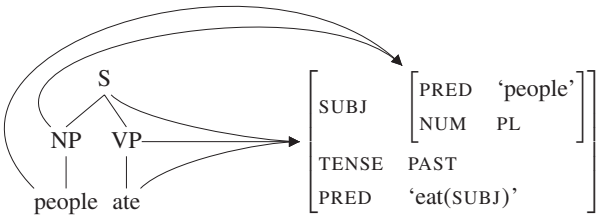

Figure 23. A representation for *People ate*

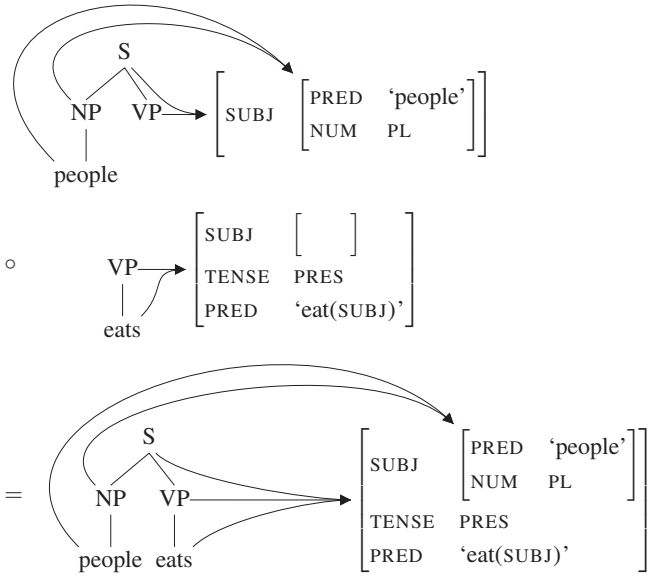

Figure 24. A valid representation for an intuitively ungrammatical sentence

This system of fragments and composition operation 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 meta-linguistic 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).

For reasons of space we will not go into the probability model of LFG-DOP, which is a relatively straightforward extension of Tree-DOP's probability model laid out in Section 2 – see Bod and Kaplan (1998, 2003).

4. Discussion: The DOP hypothesis and universal representation

We have developed an exemplar-based model of syntax that is robust in that it can parse ungrammatical input, and that offers a formal account of metalinguistic judgments at the same time. It also allows for modeling gradient judgments of acceptability by a probability function, both for grammatical and ungrammatical sentences. Thus while LFG-DOP distinguishes between grammaticality and ungrammaticality, it also acknowledges that at both sides there is a probabilistic continuum of acceptability, suggesting that the end points of least ungrammatical and least grammatical may touch.

Although there have been previous stochastic approaches to modeling acceptability judgments (see Crocker and Keller to appear for an overview), DOP distinguishes itself from these approaches by taking into account exemplars of *arbitrary* size. This allows the DOP approach to capture prefabs wherever they occur in the corpus (see Manning and Schütze 1999: 446). For example, suppose that we want to produce a sentence corresponding to a meaning of asking someone's age. There may be several sentences with such a meaning, like *How old are you?*, *What age do you have?* or even *How many years do you have?* Yet the first sentence is more acceptable than the other ones in that it corresponds to the conventional way of asking someone's age in English. This difference in acceptability is reflected by the different probabilities of these sentences in a representative corpus of English. While the probability of, e.g., *What age do you have?* is likely to be small, since it will most likely not appear as a prefabricated unit in the corpus and has to be constructed out of smaller parts, the probability of *How old are you?* is likely to be high since it can *also* be constructed by one large unit. As we showed at the end of Section 2, DOP's probability model prefers sentences that can be constructed out of the largest

possible parts from the corpus. (And even in case both sentences would occur in a representative corpus of English, *How old are you?* would have the highest frequency.)

Thus the DOP model prefers sentences and sentence-analyses that consist as much as possible of prefabs rather than "open choices". This does not mean that *What age do you have?* is less grammatical; it only stands at a lower rank of acceptability. The same kind of reasoning can be applied to sentences such as *What time is it?* versus *How late is it?*. Both sentences are grammatical with respect to a representative corpus of English, but the first sentence has a higher rank of acceptability, since the construction *What time is it?* is more probable in English than *How late is it?* for the particular meaning to be conveyed (i.e., asking the time). Interestingly, in Dutch and German, this is the other way round, as one says e.g. in Dutch *Hoe laat is 't?* rather than *Welke tijd is 't?*. Note that prefabricated word combinations are extremely frequent: Erman and Warren (2000) found that prefabs constitute at least 55 % of both spoken and written language.

The advantage of LFG-DOP over Tree-DOP is that it not only takes into account constructions and prefabs of arbitrary size, but that it also provides a linguistic account of grammaticality. In Bod (2000), LFG-DOP was tested against English native speakers who had to decide as quickly as possible whether three-word (subject-verb-object) sentences were grammatical. The test sentences were selected from the British National Corpus (BNC) and consisted of both frequent sentences such as *I like it* and low-frequency sentences such as *I keep it*, as well as a number of ungrammatical pseudo-sentences. This pilot experiment showed that frequent sentences are recognized more easily and quickly than infrequent sentences, even after controlling for plausibility, word frequency, word complexity and syntactic structure. Next, a simple implementation of LFG-DOP was used to parse the test sentences. Each fragment f was assigned a response latency by its frequency $freq(f)$ in the BNC as $\frac{1}{\left(1+logfreq(f)\right)}$ – see Baayen et al. (1997). The latency of the total sentence was estimated as the sum of the latencies of the fragments. The resulting model matched very well with the experimentally obtained reaction times (up to a constant) but only if *all* fragments were taken into account. The match significantly deteriorated if two-word and three-word chunks were deleted.

The result of this experiment triggers the hypothesis that "the accuracy increases with increasing fragment size". This hypothesis has been corroborated not only for three-word sentences, but also for the much longer sentences from the Penn Treebank. The term "accuracy" should be broadly interpreted here: it can refer to the accuracy of DOP's prediction of human reaction times, but also to the accuracy of predicting the parse tree of an utterance as annotated by humans in the Penn Treebank (Bod 1998, 2001), and even to the accuracy of

translating a sentence from one language into another (Hearne and Way 2003). For all these domains the hypothesis has been corroborated. This led me to conjecture in Bod (2001) that "the increase in accuracy with increasing fragment size" is independent of the language, referring to it as the *DOP hypothesis*. This hypothesis is a risky generalization which can be refuted by one reliable counter-example. Yet, to-date, the DOP hypothesis has been successfully tested for several languages, such as English, Dutch and French (Bod 1998; Hearne and Way 2003; Cormons 1999), Hebrew (Sima'an et al. 2001) and Mandarin (Hearne and Way 2004). The hypothesis has also been corroborated for different linguistic representations, from LFG to HPSG and TAG (cf. Bod et al. 2003a).

However, these experiments are all carried out on rather unrealistic corpora that do not correspond to a person's memory of previous language experiences. The only *cognitively* interesting corpus would be a collection of all language utterances ever heard and uttered in a language user's experience. Not even the *Childes* corpus (MacWhinney 1995) comes close to this, as it is still based on a mix of different users (and all users are children and their caregivers). The development of cognitively realistic corpora will be one of the main future goals in exemplar-based syntax.

Of course we should again raise the question where the initial representations come from. In Section 1, we argued that tree structures can be induced from distributional statistics (Bod 2006). How can we extend such a bootstrapping approach to richer representations like functional categories and argument structures? It turns out that the induction of what are usually called "richer" representations is easier than the induction of "simple" surface constituent trees. For example, Gildea (2002) and Swier and Stevenson (2004) show how grammatical functions, verb-argument structures and semantic roles can be bootstrapped using an iteratively updating probability model. And Bod (2006) discusses how trees can be initially enriched with all possible combinations of grammatical functions and argument structures using DOP's probability model to decide which of these combinations are most useful in parsing new data. While the details of these bootstrapping models fall beyond the scope of this article, it is important to note that they can only learn grammatical functions and semantic roles for new input if there is a definition of these functions and roles to begin with. In other words, these models rely on an a priori definition of what we called a *well-formed representation* of a linguistic utterance.

While the need for such a definition is sometimes seen as a limitation of bootstrapping methods, we believe it is a fundamental prerequisite for any exemplar-based model of language. In principle, such a definition should apply to all linguistic phenomena and all languages and may thus be referred to as *Universal Representation*. If there is anything innate in the human language faculty it should be this data structure by which linguistic utterances are repre-

sented. The notion of Universal Representation may seem similar to the notion of Universal Grammar in generativist linguistics, but there is one very important difference: given a definition of a well-formed representation, the assignment of representations to language utterances is accomplished solely on the basis of statistics – both in language acquisition and adult language processing. The notion of grammatical rule may still appear in the scientific discourse, but is not part of a native speaker's competence. The only "rules" in exemplar-based syntax are the decomposition and recomposition rules that construct new representations out of previous representations. On this account, knowledge of language is viewed not as a grammar but as a statistical ensemble of language experiences that slightly changes every time a new utterance is perceived or produced.

University of St Andrews

References

Abeillé, Anne (2003). *Treebanks*. Dordrecht: Kluwer Academic Publishers.

Baayen, Harald, Ton Dijkstra and Rob Schreuder (1997). Singulars and plurals in Dutch: Evidence for a parallel dual route model. *Journal of Memory and Language* 36: 94–117.

Barlow, Michael and Susan Kemmer (2000). *Usage-Based Models of Language*. Stanford: CSLI Publications.

Bock, Kathryn (1995). Sentence production: From mind to mouth. In *Handbook of Perception and Cognition*, vol. xi: *Speech, Language and Communication,* J. Miller and P. Eimas (eds.), 181–216. Orlando: Academic Press.

Bock, Kathryn and Helga Loebell (1990). Framing sentences. *Cognition* 35: 1–39.

Bod, Rens (1992). Data-oriented parsing. In *Proceedings COLING'92*, C. Boitet (ed.), 855–859. Nantes: ICCL.

— (1998). *Beyond Grammar: An Experience-Based Theory of Language.* Stanford: CSLI Publications.

— (1999). Context-sensitive spoken dialogue processing with the DOP model. *Journal of Natural Language Engineering* 5 (4): 306–323.

— (2000). The storage and computation of three-word sentences. Paper presented at *AMLaP' (2000)*. Leiden.

— (2001). What is the minimal set of fragments that obtains maximum parse accuracy? *Proceedings ACL'(2001)*, Toulouse, 66–73.

— (2002). A unified model of structural organization in language and music, *Journal of Artificial Intelligence Research* 17: 289–308.

— (2003a). Introduction to elementary probability theory and formal stochastic language theory. In *Probabilistic Linguistics*, Rens Bod, Jennifer Hay and Stefanie Jannedy (eds.), 11–37. Cambridge: The MIT Press.

— (2003b). An efficient implementation of a new DOP model. *Proceedings EACL'03*, Budapest, 28–35.

— (2005). Combining supervised and unsupervised natural language processing. Paper presented at *CLIN'05*, Amsterdam.

— (2006). An all-subtrees approach to unsupervised parsing. *Proceedings COLING-ACL 2006*, Sydney, 865–872.

Bod, Rens and Ronald Kaplan (1998). A probabilistic corpus-driven model for lexical functional analysis, *Proceedings COLING-ACL'98*, Montreal, 145–151.

— (2003). A DOP model for lexical-functional grammar. In *Data-Oriented Parsing*, Rens Bod, Remko Scha and Khalil Sima'an (eds.), 211–232. Stanford: CSLI Publications.

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

Bresnan, Joan (2000). *Lexical-Functional Syntax*, Oxford: Blackwell.

Bybee, Joan (to appear). From usage to grammar: The mind's response to repetition. *Language* 82.

Cormons, Boris (1999). Analyse et desambiguisation: une approche a base de corpus (data-oriented parsing) pour les representations lexicales fonctionelles. PhD dissertation. University of Rennes.

Crocker, Matthew and Frank Keller (to appear). Probabilistic grammars as models of gradience in language processing. In *Gradience in Grammar: Generative Perspectives*, G. Fanselow, C. Féry, R. Vogel and M. Schlesewsky (eds.). Oxford: Oxford University Press.

Daelemans, Walter and Antal van den Bosch (2005). *Memory-Based Language Processing.* Cambridge: Cambridge University Press.

Erman, Britt and Beatrice Warren (2000). The idiom principle and the open choice principle. *Text* 20 (1): 29–62.

Fillmore, Charles, Paul Kay and Mary O'Connor (1988). Regularity and idiomaticity in grammatical constructions: The case of *let alone*. *Language* 64: 501–538.

Gahl, Susanne and Susan Garnsey (2004). Knowledge of grammar, knowledge of usage: Syntactic probabilities affect pronunciation variation. *Language* 80 (4): 748–775.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum and Ivan Sag (1985). *Generalized Phrase-Structure Grammar.* Harvard: Blackwell and Harvard University Press.

Giere, Ronald (1988). *Explaining Science*. Chicago: The University of Chicago Press.

Gildea, Daniel (2002). Probabilistic models of verb-argument structure *Proceedings COLING'02*, Taipei, 308–314.

Goldberg, Adele (1995). *Constructions*. Chicago: University of Chicago Press.

— (2006). *Constructions at Work: The Nature of Generalization in Language.* Oxford: Oxford University Press.

Goodman, Joshua (2003). Efficient parsing of DOP with PCFG-reductions. In *Data-Oriented Parsing*, Rens Bod, Remko Scha and Khalil Sima'an (eds.), 125–146. Stanford: CSLI Publications.

Hearne, Mary and Andy Way (2003). Seeing the wood for the trees: data-oriented translation. *Proceedings of MT Summit IX*, 165–172. New Orleans.

— (2004). Data-oriented parsing and the Penn Chinese treebank. *Proceedings First International. Joint Conference on Natural Language Processing*, 406–413. Hainan Island.

Jurafsky, Daniel (2003). Probabilistic modeling in psycholinguistics: linguistic comprehension and production. In *Probabilistic Linguistics*, Rens Bod, Jennifer Hay and Stefanie Jannedy (eds.), 39–95. Cambridge: The MIT Press.

Kaplan, Ronald (1989). The formal architecture of lexical-functional grammar. *Journal of Information Science and Engineering* 5: 305–322.

Kaplan, Ronald and Joan Bresnan (1982). Lexical-functional grammar: a formal system for grammatical representation. In *The Mental Representation of Grammatical Relations*, Joan Bresnan (ed.), 173–281. Cambridge: The MIT Press.

Klein, Dan and Manning, Chris (2005). Natural language grammar induction with a generative constituent-context model. *Pattern Recognition* 38: 1407–1419.

Kuhn, Thomas (1970). *The Structure of Scientific Revolutions*, 2nd edition, Chicago: University of Chicago Press.

MacWhinney, Brian (1995). *The CHILDES project*. Mahwah: Lawrence Erlbaum Associates.

Manning, Chris (2003). Probabilistic syntax. In *Probabilistic Linguistics*, Rens Bod, Jennifer Hay and Stefanie Jannedy (eds.), 289–341. Cambridge: The MIT Press.

Manning, Chris and Hinrich Schütze (1999). *Foundations of Statistical Natural Language Processing*. Cambridge: The MIT Press.

Marcus, Mitchell, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Anne Bies, Mark Ferguson, Karen Katz and Britta Schasberger (1994). The Penn treebank: Annotating predicate argument structure. *Proceedings ARPA Human Language Technology Workshop*, 114–119. Menlo Park: Morgan Kaufmann.

Nickles, Thomas (2003). Normal science: From logic to case-based and model-based reasoning. In *Thomas Kuhn*, Nickles, T. (ed.), 142–177. Cambridge: Cambridge University Press.

Nosofsky, Robert (1988). Similarity, frequency and category representation. *Journal of Experimental Psychology: Learning, Memory and Cognition* 14: 54–65.

Pierrehumbert, Janet (2001). Exemplar dynamics: Word frequency, lenition and contrast. In *Frequency and the Emergence of Linguistic Structure*, Joan Bybee and Paul Hopper (eds.), 137–158. Amsterdam: John Benjamins.

Sag, Ivan, Thomas Wasow and Emily Bender (2003). *Syntactic Theory: A Formal Introduction*. Stanford: CSLI Publications.

Scha, Remko (1990). Taaltheorie en taaltechnologie; competence en performance. In *Computertoepassingen in de Neerlandistiek*, Q. de Kort and G. Leerdam (eds.), 7–22. Almere: LVVN-jaarboek.

Sima'an, Khalil, Alon Itai, Yoad Winter, Alon Altman and Noa Nativ (2001). Building a treebank of modern Hebrew text. *Journal Traitement Automatique des Langues. Special Issue on Natural Language Processing and Corpus Linguistics* 42 (2): 347–380.

Skousen, Royal (1992). *Analogy and Structure*. Dordrecht: Kluwer Academic Publishers.

Swier, Robert and Suzanne Stevenson (2004). Unsupervised semantic role labeling. *Proceedings EMNLP'04*, 95–102. Barcelona.

Tomasello, Michael (2003). *Constructing a Language*. Harvard: Harvard University Press.

Van Zaanen, Menno (2001). Bootstrapping structure into language: Alignment-based learning. PhD Dissertation, University of Leeds.

Verhagen, Arie (2005). *Constructions of Intersubjectivity*. Oxford: Oxford University Press.

Zollmann, Andreas and Khalil Sima'an (2005). A consistent and efficient estimator for data-oriented parsing. *Journal of Automata, Languages and Combinatorics* 10: 367–388.