[Published in: *Journal of Experimental and Theoretical Artificial Intelligence*, **11**(1999): 441-471]

A Hybrid Architecture for Robust MT using LFG-DOP

Andy Way,
School of Computer Applications,
Dublin City University,
Dublin 9, Ireland.

Tel: +353-1-7045644
Fax: +353-1-7045442
Email: away@compapp.dcu.ie

May 24, 1999

Abstract

We develop a model for Machine Translation (MT) based on Data-Oriented Parsing (DOP) allied to the syntactic representations of Lexical Functional Grammar (LFG).

We begin by showing that in themselves, none of the main paradigmatic approaches to MT currently suffice to the standard required. Nevertheless, each of these approaches contains elements which if properly harnessed should lead to an overall improvement in translation performance. It is in this new hybrid spirit that our search for a better solution to the problems of MT can be seen.

We summarise the original DOP model (Bod 1992), as well as the DOT model of translation on which it is based (Poutsma 1998). We demonstrate that DOT is not guaranteed to produce the correct translation, despite provably deriving the most probable translation. We go on to critically evaluate previous attempts at LFG-MT, commenting briefly on particular problem cases for such systems. We then show how the LFG-DOP model of Bod & Kaplan (1998) can be extended to serve as a novel hybrid model for MT which promises to improve upon DOT, as well as the pure LFG-based translation model.

1 Introduction

A welcome recent development in the field of MT is the recognition that none of the three main paradigmatic approaches to MT, namely:

- Transfer-based, e.g. Eurotra, (Arnold & des Tombe, 1987); METAL, (Bennett & Slocum, 1985)
- Interlingua-based, e.g. KBMT, (Goodman & Nirenburg, 1991); Rosetta, (Landsbergen, 1989; Rosetta, 1994)
- Statistics-based, e.g. IBM-MT, (Brown *et al.*, 1990; 1992a)

in themselves perform the task of fully-automated, high quality translation to the standard required.

1.1 Rule-based Approaches (Transfer & Interlingua)

The biggest single problem with rule-based (primarily transfer) systems is that of *knowledge acquisition*. This takes several forms (Su & Chang, 1992, p.255):

- Wide *coverage* of texts is difficult to achieve. Knowledge is (normally) restricted to (theoretically interesting) interactions of linguistic phenomena (whether these occur frequently or not in real corpora is deemed irrelevant). Consequently expansion from dealing with 'toy' grammars and lexica often leads to lack of robustness.
- Given that the translation data is (often) invented (rather than extracted from real corpora), it is difficult to maintain *consistency* in the knowledge bases between developers.
- Approaches are (often) based on existing linguistic theories, which are themselves incomplete. Therefore, it is tempting to resort to *ad hoc* procedures when faced with constructions not dealt with in the theory.
- They find it hard to deal with *ill-formed input*. Again, given the fact that they are often based on grammatical theories, most only accept well-formed strings. Real text, unfortunately, is not always so accommodating.

Other criticisms having similar roots include:

- Expanding one's coverage may cause newly added rules to impinge on others in an unpredictable fashion, causing previously correct behaviour to be inadvertently undone. This might be termed a problem of *tuning*.
- There is (normally) no systematic basis to the *acquisition of rules*, so that while being of theoretical interest such systems may be of little real relevance.
- It is difficult to handle *uncertainty*, i.e. if such systems incorporate a *preference* mechanism, this (normally) has no empirical objectivity or consistency underpinning it.

Given this, one might question why this paradigm has proved the most popular over the years with MT developers. Of course, there are a number of advantages to transfer-based approaches, although one must bear in mind that many of these so-called advantages are in fact 'non-disadvantages' inherent in other approaches, notably interlingua-based systems.

For the purposes of this discussion we will assume the following applies equally to interlingual and knowledge-based MT (KBMT). Such approaches are viewed as attractive in theory, but unattainable in practice. Truly language neutral interlinguae are unachievable, and indeed, for closely related languages it makes little sense to ignore their similarities in translation. Consequently, a more pragmatic approach is usually taken which stops shy of such an intellectually appealing, but ultimately impractical stipulation that the intermediate representation be language-*neutral*, accepting instead that they be merely language-*independent* [1]. Note that the previous criticism still stands: it makes little sense to ignore the similarities of (say) Spanish and Portuguese when translating between them, when making use of this shared information would prove more fruitful.

Nirenburg *et al.* (1992, p.51) term this the 'maximalist' view of interlingua, and discuss it among a plethora of oft-cited criticisms of such approaches, focusing particularly on arguments for and against 'meaning-oriented' MT. One of the criticisms (*op cit.*, p.43) is that meaning is not *required* for translation, a view with which advocates of statistics-based methods would agree (e.g. Brown *et al.*, 1990, 1992a). Nirenburg *et al.* (*op cit.*, p.46) state, however, that the processes involved in the statistical approach, particularly 'viewing language not as a productive system but as a fixed set of canned locutions ... moves MT out of applied science and into pure engineering'; not that there is anything wrong with this *per se*, of course. They continue:

> 'Completely uninterpreted comparison (of text corpora) will lead to errors simply because the human translators who produced the translations in the corpus in the first place do not translate word-for-word or even sentence-for-sentence.' (Nirenburg *et al.* (*op cit.*, p.46)

While there is little doubt that this is true, we intend to show that a hybrid approach combining a DOP treebank together with the linguistic structures provided by Lexical Functional Grammar (LFG) ought to produce fewer such errors, leading to an overall improvement in translation output, both in terms of quality and robustness.

One of the major criticisms of rule-based MT (RBMT) concerns generation of the target language, where transfer-based approaches tend to preserve the syntactic structure of the source text in translation, so far as this is possible. As Nirenburg *et al.* point out (*op cit.*, p.55):

> 'Direct structural correspondences between certain pairs of languages can be exploited in MT systems of a particular type, but they should be treated as idiosyncratic occasions rather than phenomena that occur as a rule .... However, if an MT system does not possess sufficient knowledge to analyze source language texts deeply enough ... it may rely on preserving the syntax of the source text in the target text as a very crude default heuristic'.

Such criticisms have been taken into account in systems as diverse as Shake & Bake MT (Whitelock 1992; Beaven 1992) and Statistical MT (Brown *et al.*, 1990; 1992a), where the target strings are produced from a number of target words in a 'bag' according solely to rules in the target grammar, and with no reference to the source string at all. We shall see that our proposed model for LFG-DOP MT similarly avoids this criticism by producing target strings from target LFG f-structures.

1.2 Statistical Approaches

In their seminal work on statistical approaches to MT, Brown *et al.* (1990) claim that linguistic information may be dispensed with entirely. However, in subsequent papers (Brown *et al.*, 1992a), they dampen down this claim in recognising the need to incorporate low-level linguistic information (although done here in a piecemeal way) in order to capture morphological variants of each word to improve their statistical model.

Nevertheless, compared with rule-based approaches, 'pure' statistical methods assume no linguistic models (of course), nor do they have 'any methods of strict well-formedness in mind' (Su & Chang, 1992, p.250). Consequently, it is possible to cope with ill-formed input using such approaches [2], while not being tied to any linguistic theory enables easy computation (cf. Brown *et al.* 1990). Other advantages of this type of approach include (Su & Chang 1992, p.252):

- Uncertainty is interpreted objectively.
- Consistency is maintainable even in large-scale systems.
- System parameters can be manipulated language-independently.
- Training is possible with little human intervention.

In addition, it might be argued that a good deal of the knowledge needed for MT is *inductive*, rather than *deductive*, in that while linguistics is induced from languages, no *natural* language is generated from any linguistic theory *per se*. All these facts would argue in favour of a statistics-based approach. However, there are problems too, as one might expect:

- The statistical approach requires huge, good quality bilingual (or multilingual) corpora to be available. This is currently the case for few languages only, rendering this approach to MT rather limited.
- Of course, if the corpora are too small, one faces the problem of *sparse data*, where one's statistical models could prove unreliable.
- They also need to be *representative*, for word frequency strongly depends on the domain and text type.
- Given the lack of any linguistic knowledge, the parameter space is normally impractically large. One tends, therefore, to sacrifice the quality of the statistical information (using bigram models rather than trigrams, for instance) at the expense of functionality (*sparseness*, again).

In addition, almost all statistical approaches can deal only with *local phenomena*, i.e. there are certain constructions (e.g. long-distance dependencies), which cannot be dealt with by most such methods. Furthermore, it would appear that the effect of *distortion*, i.e. how closely aligned words are between the two languages (e.g. allowing adjectives to appear after their nouns in French, but before them in English), may cause one to question the effectiveness of such an approach between two languages whose surface orders are not as closely mirrored (English and Japanese, say). Even between closely related languages, one can foresee problems: it is non-trivial to gather accurate statistics to find correlations between English and German verbs, for instance, which, in complex sentences at least, appear in rather different surface positions.

1.3 Example-based Methods

Much of the above remains relevant for Example-based methods (EBMT — also known as Memory-Based Translation). For instance, there are extreme adherents to this approach also, who suggest that EBMT should deal with the whole process of translation (Sato & Nagao, 1990). There are other more moderate proposals which prefer instead to combine this approach with others (e.g. Sumita *et al.*, 1990). We shall show that at a shallow level, DOP-based models can serve as exemplars of EBMT, but taking better advantage of DOP (and LFG-DOP) models allows a fully fledged MT system to be developed in its own right.

1.4 Hybrid Approaches

It should now be clear that dogmatic adherence to one methodology will result in sub-optimal results. Indeed, at least with reference to the claim that statistics-based techniques suffice, this is unnecessary since many regular aspects of language can be handled quite simply using rules. At the same time, it is clear that each of the approaches contains favourable elements which, if integrated into a single system, could do the job better than any of the distinct methods, a view endorsed by many other researchers (cf. Carbonnel *et al.*, 1992:235; Lehmann & Ott, 1992:237; Grishman & Kosaka, 1992:263).

2 LFG-DOP: A Hybrid Architecture for NLP

Bod & Kaplan (1998) have recently augmented DOP with the syntactic representations of Lexical Functional Grammar (LFG) in the spirit of this trend towards hybridity. We propose here that LFG-DOP be utilised as a basis for machine translation (MT). This will be described in detail later in the paper, but it is first necessary to outline both the original DOP model as well as how LFG relates to MT, as these provide the theoretical backbone to the proposed system.

2.1 The DOP Approach

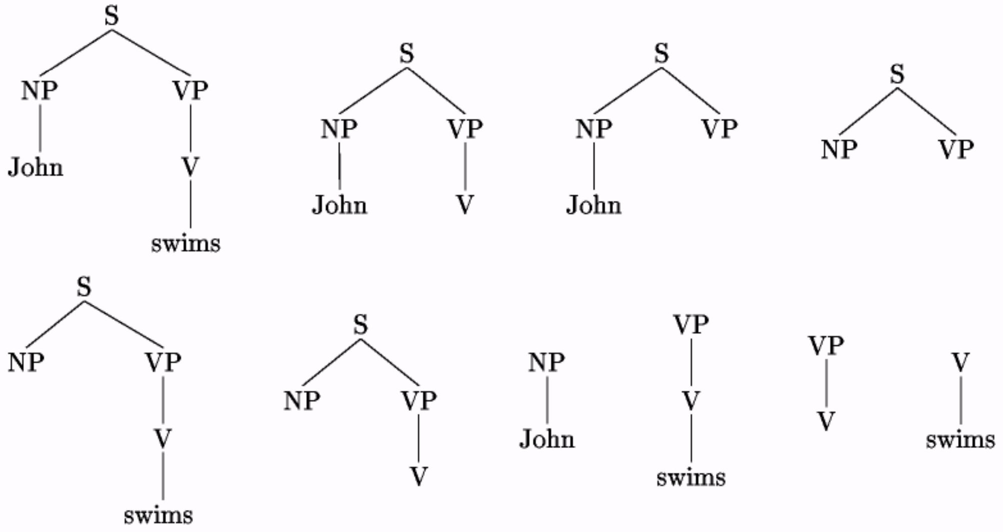
Data-Oriented Parsing (DOP) language models (e.g. Bod 1992, 1993, 1995; Sima'an 1995; Rajman 1995) assume that past experiences of language are significant in both perception and production. DOP prefers performance models over competence grammars, in that abstract grammar rules are eschewed in favour of models based on large collections of previously occurring fragments of language. New language fragments are processed with reference to already existing fragments from the treebank, which are combined using probabilistic techniques to determine the most likely analysis for the new fragment.

The general DOP architecture Bod (1995) stipulates four parameters on which particular models are instantiated, namely:

1. A formal definition of a well-formed representation for fragment analyses.
2. A set of decomposition operations for splitting strings into a set of fragments.
3. A set of composition operations for recombination of such fragments in order to derive analyses of new strings.
4. A definition of a probability model indicating the likelihood of a string based on the probabilities of its constituent parts.

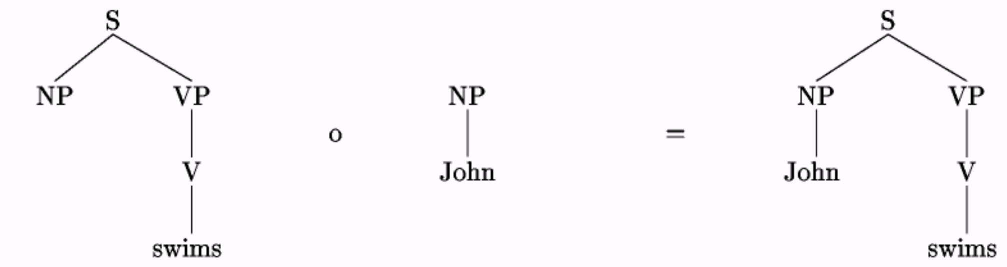
DOP models typically use surface PS-trees as the chosen representation for strings (hence 'Tree-DOP', Bod 1992), but nothing hangs on this choice. However, given that LFG c-structures are little more than annotated PS-trees allows us to proceed very much on the same lines as in Tree-DOP, which has two decomposition operations to produce subtrees from sentence representations: (i) the *Root* operation, which takes any node in a tree as the root of a new subtree, deleting all other nodes except this new root and all nodes dominated by it; and (ii) the *Frontier* operation, which selects a (possibly empty) set of nodes in the newly created subtree, excluding the root, and deletes all subtrees dominated by these selected nodes. For instance, the full set of DOP trees derived from the sentence *John swims* would be as in (1):

(1)



Tree-DOP recombines fragments starting from the leftmost non-terminal frontier node, and replaces this with a fragment having the same root symbol as this frontier node. For instance, assuming the treebank in (1), *John swims* has (2) as a possible derivation (among many others):

(2)



Finally, the chosen probability model for Tree-DOP is based quite simply on the relative frequencies of fragments in the treebank, assuming (i) that the trees are stochastically independent; and (ii) that the treebank in question represents the total population of subtrees. Of course, neither of these is correct, but their adoption allows both the construction of a simple probability model as well as easy subsequent computation.

These elements enable representations of new strings to be constructed from previously occurring fragments in a number of ways. If each derivation D has a probability $\mathrm{P}(D)$, then the probability of deriving a Tree-DOP representation R is the sum of the probabilities of the individual derivations, as in (3):

(3)

$$P(R) = \sum D\ derives\ R\ P(D)$$

The probability of each individual derivation t is calculated as the product of the probabilities of all the constituent elements ($\langle t_1, t_2 ... t_n \rangle$, here) involved in choosing tree t from the treebank, as in (4):

(4)

$$P(\langle t_1, t_2 ... t_n \rangle) = \prod_{i=1}^{n} \frac{P(t)}{\sum_{t' \in corpus} P(t')}$$

This is best illustrated with a simple example. Given two sentences—*John swims* and *Peter laughs*—and their associated trees, we shall derive the probability of the new string *Peter swims* with respect to this small corpus of tree fragments. This is the joint probability (i.e. the product of the individual probabilities) of:

1. selecting the subtree **s[NP vp[v[swims]]]** among the subtrees labelled S.

2. selecting the subtree **np[Peter]** among the subtrees labelled NP.

i.e. P(t = [**NP vp[v[swims]]**] | root(t) = S) * P(t = [**np[Peter]**]| root(t) = NP) These conditional probabilities are computed by dividing the cardinalities of the occurrences of the trees. For instance, P(t = **np[Peter]** | root(t) = NP) =

$$\frac{\#(\mathbf{np[Peter]} \mid root(t) = NP)}{\#(t \mid root(t) = NP)}$$

Given this small corpus, P(t = [**NP vp[v[swims]]**] | root(t) = S) = 1/12, i.e. there are 12 trees possible with *Root* = *S*, shown in (5):

(5) a. s(np(john),vp(v(swims)))

b. s(np(john),vp(v(Y)))

c. s(np(john),vp(Y))

d. s(np(X),vp(v(swims)))

e. s(np(X),vp(v(Y)))

f. s(np(X),vp(Y))

g. s(np(peter),vp(v(laughs)))

h. s(np(peter),vp(v(Y)))

i. s(np(peter),vp(Y))

j. s(np(X),vp(v(laughs)))

k. s(np(X),vp(v(Y)))

l. s(np(X),vp(Y))

only one of which (5d) matches this structure. The copies of trees are produced in different ways: for instance, the tree pattern `s(np(X),vp(Y))`, seen as (5f) and (5l) here, is a subtree of the full parse tree of both example sentences (the variables are so that Prolog can later instantiate them to actual pieces of structure in the recombination process—in a visual representation of such trees, as in (1), we can safely omit them, of course). Importantly, then, we can see that a DOP treebank is a bag, rather than a set. Each tree which can play a part in combining together with other trees to form a representation for a sentence is used to contribute to the overall probability of that representation given the corpus.

It is not difficult to see that P(t = [**np[Peter]**]| root(t) = NP) = 1/2, as there are 2 trees possible with *Root* = *NP*, one for *Peter* and one for *John*. As a result, therefore, P(*Peter swims*) = 1/12 * 1/2 = 1/24, assuming this derivation. This probability is small, and does not reflect the probability of the string *Peter swims* given this corpus, as the probability of a parse-tree is computed by considering all its derivations, as in (3),which can also be written as (6):

(6)

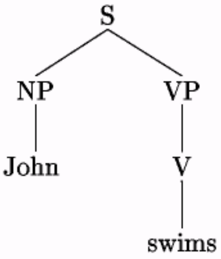
$$\sum{}^{i} \prod{}^{j} \frac{\#(t_{ij})}{\#(t \mid root(t) = root(t_{ij}))}$$

There are 7 such derivations possible, giving P*(Peter swims*) = 40/196 = 5/24, reflecting the fact that of the 4 possible sentences derivable using this toy corpus, the original sentences *John swims* and *Peter laughs* are more probable (7/24 each) than the new derivable strings *Peter swims* and *John laughs*. The reason for this, of course, is that full trees exist for the original strings (with probabilities 1/12 each). DOP finds these as well as the remaining trees labelled S used to derive the other two sentences, hence the difference of 2/24, or 1/12. Despite the triviality of the small corpus chosen here, this illustration gives the reader a flavour of the simplicity of the DOP approach given such a probability model, as well as the preference in DOP (in terms of higher probability) for larger trees. We shall take advantage of this later when it comes to translation, as the same effects are seen there too.

2.2 Approaches to Translation using Lexical Functional Grammar

LFG (Kaplan & Bresnan, 1982) is theory of syntax which allows description of sentences at a number of different levels—c-structure, f-structure and s-structure. The c-structure (*constituent* structure) is a phrase-structure tree signifying the surface structure of the string, the f-structure (*functional* structure) is an attribute-value matrix (AVM) capturing the grammatical relations inherent in the string, while s-structures (*semantic* structure) go one level deeper and express relational information. There are two mappings—ϕ, which maps c-structure nodes onto elements of the f-structure, and σ, which relates f- and s-structures, so that LFG is a linear model (c $\longrightarrow$f $\longrightarrow$s-structure). An example of the three structures for the sentence *John swims* is shown in (7):

(7)



$$\xrightarrow{\phi} \begin{bmatrix} \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{'John'}\\ \text{NUM} & \text{SG}\end{bmatrix} \\ \text{PRED} & \text{'swim}\langle(\uparrow\text{SUBJ})\rangle\text{'} \\ \text{TENSE} & \text{PRES}\end{bmatrix} \xrightarrow{\sigma} \begin{bmatrix}\text{REL} & \text{SWIM}\\ \text{ARG1} & \text{JOHN}\end{bmatrix}$$

LFG has also been proposed as an MT formalism (Kaplan *et al.* 1989). Given the very powerful and elegant way of relating unlike representations (c-structure trees and f-structure AVMs) when used as a pure-linguistics theory of syntax, it is not too surprising that its use as an MT engine has met with some success. Kaplan *et al.* illustrate the ability of LFG to cope with some hard examples using *codescription* i.e. various levels of linguistic structure contribute to ('co-describe') the translation of strings. Rather than conflating all translationally relevant information into a single, linguistically hybrid level of representation, LFG-MT allows information from different linguistic levels of representation to interact in order to constrain the translation relation, by function composition.

The primary example used here by Kaplan *et al.* is the well-known headswitching example *venir de X* ⇔ *has just X-ed*), which their LFG-MT system proposes to deal with as in (8):

(8) *just*:
$(\uparrow \text{PRED}\ ) = \text{'just}\langle\uparrow \text{ARG}\ \rangle\text{'}$
$(\tau\uparrow \text{PRED FN}) = \text{venir}$
$(\tau\uparrow \text{XCOMP}) = \tau(\uparrow \text{ARG})$

That is, the XCOMP function (i.e. infinitival complement) of *venir* corresponds to the ARG function of *just*.

Nevertheless, Sadler *et al.* (1989, 1990) show that this approach cannot deal elegantly and straightforwardly with more complex cases of headswitching, as in (9):

(9) a. Jan zwemt toevallig graag ⟶

b. Jan happens to like to swim

c. Jan likes to happen to swim

The f-structure corresponding to the Dutch sentence (9a) is (10):

$$(10)\quad \left[\begin{array}{ll} \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`Jan'}\\ \text{NUM} & \text{SG}\end{array}\right] \\ \text{PRED} & \text{`zwemmen}\langle(\uparrow\text{SUBJ})\rangle\text{'} \\ \text{TENSE} & \text{PRES}\\ \text{ADJUNCT} & \left[\ \text{toevallig, graag}\ \right]\end{array}\right]$$

Given the original formulation of LFG, there is no way of producing the required embedding of *graag* ('likingly') under *toevallig* ('by chance'), and not vice versa, without resorting to tuning. Furthermore, satisfying the requirement that only possible translations are produced is problematic where the translation of a lexical head is conditioned in some way by its dependants, as in *commit suicide* ⟶*se suicider*, cf. (26-28), (30).

Cases with adverbial modifiers like (9) are subsequently solved by the introduction of the notion of *restriction*, which seeks to overcome problems in mapping between flat syntactic f-structures to hierarchical semantic ones (Kaplan & Wedekind 1993). The intuition is that in such cases semantic units correspond to subsets of functional information, and restricting the f-structure (in other words, removing *graag* and *toevallig* in turn from the adjunct set in (10)) enables (10) to be associated with the alternative s-structures in (11):

$$(11)\quad \left[\begin{array}{ll} \text{REL} & \text{`toevallig'} \\ \text{ARG1} & \left[\begin{array}{ll}\text{REL} & \text{`graag'}\\ \text{ARG1} & \left[\begin{array}{ll}\text{REL} & \text{`zwemmen'}\\ \text{ARG1} & \text{`Jan'}\end{array}\right]\end{array}\right]\end{array}\right] \left[\begin{array}{ll} \text{REL} & \text{`graag'} \\ \text{ARG1} & \left[\begin{array}{ll}\text{REL} & \text{`toevallig'}\\ \text{ARG1} & \left[\begin{array}{ll}\text{REL} & \text{`zwemmen'}\\ \text{ARG1} & \text{`Jan'}\end{array}\right]\end{array}\right]\end{array}\right]$$

This addition to LFG-MT, while solving most of the immediate problems, has met with other criticisms, notably those of Butt (1994), who states that the use of the restriction operator entails dealing with complex predicates in Urdu in an unintuitive way. Butt consequently advocates the use of linear logic source and target semantic representations (Dalrymple *et al.*, 1993) as a more flexible solution to this problem, in conjunction with the classical τ-equations and language specific (i.e. monolingual) mapping principles between f- and s-structures. The use of linear logic has also recently been adopted by Van Genabith *et al.* (1998), who recommend it as a formalism for the representation of transfer, in addition to performing transfer on linear logic meaning constructors.

In sum, the firm mathematical foundations of LFG-MT provide the linguist with a powerful yet intuitive set of tools which facilitate the description of natural languages both monolingually and multilingually. Nevertheless, it suffers, as all rule-based systems do, from a lack of robustness. It is this shortcoming that its alliance with DOP will help overcome.

2.3 Opportunities for Hybridity-LFG DOP

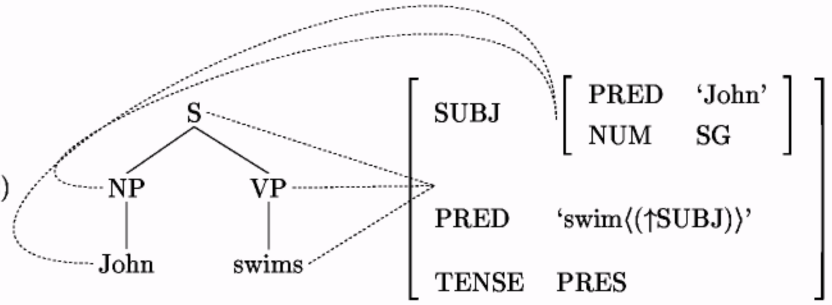
Tree-DOP has been used to perform experiments on the Penn Treebank [3] and the OVIS [4] (Dutch Public Transport Information System) corpus (Bod 1993, 1995; Bonnema *et al.*, 1997; Sima'an 1995) which show an increase in parse accuracy when larger tree fragments are considered. Nevertheless, such approaches are necessarily limited to those contextual dependencies actually occurring in the corpus, which is a reflection of surface phenomena only. It has been known for some time that purely context-free models are insufficiently powerful to deal with all aspects of human language. In this regard, DOP models have been augmented (van den Berg *et al.*, 1994; Tugwell 1995) to deal with richer representations, but such models have remained context-free. LFG, however, is known to be beyond context-free. It can capture and provide representations of linguistic phenomena other than those occurring at surface structure. The functional structures of LFG have been allied to the techniques of DOP to create a new model, LFG-DOP (Bod & Kaplan, 1998), which adds a measure of robustness (both with respect to unseen as well as ill-formed input) not available to models based solely on LFG.

For example, Bod & Kaplan (*op cit.*) show that given the treebank for sentences *John fell* and *people walked*, models can be constructed where, for two new strings *John walked* and *people fell*, the unmarked interpretation is less likely than the two specific interpretations, and of these the intuitively correct ones are selected for each corresponding verb. That is, if we already have a plural verb, and we come across a subject NP we do not recognise, at least we prefer it to be plural over the unspecified or singular alternatives, and vice versa.

As with Tree-DOP, LFG-DOP needs to be defined using the same four parameters outlined in §2.1. Its **representations** are simply lifted *en bloc* from LFG theory, so that each string is annotated with a c-structure, an f-structure, and a mapping ϕ between them. Well-formedness conditions operate solely on f-structure, as usual.

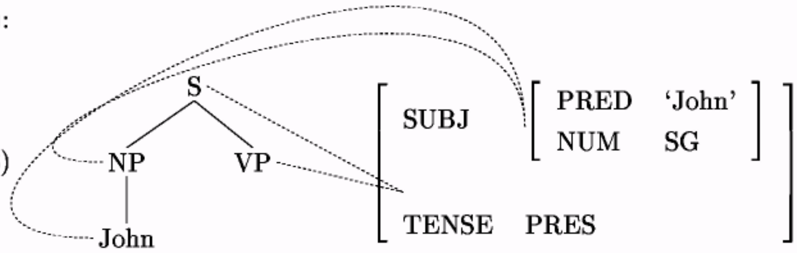
Since we are now dealing with $\langle$c,f$\rangle$ pairs of structure, the *Root* and *Frontier* **decomposition** operations of DOP need to be adapted to stipulate exactly which c-structure nodes are linked to which f-structure components, thereby maintaining the fundamentals of c- and f-structure correspondence. As in Tree-DOP, *Root* erases all nodes outside of the selected node, and in addition deletes all ϕ-links leaving the erased nodes, as well as all f-structure units that are not ϕ-accessible from the remaining nodes, reflecting the intuitive notion that nodes in a tree carry information only about the f-structure elements to which the root node of the tree permits access, as in (12):

(12)



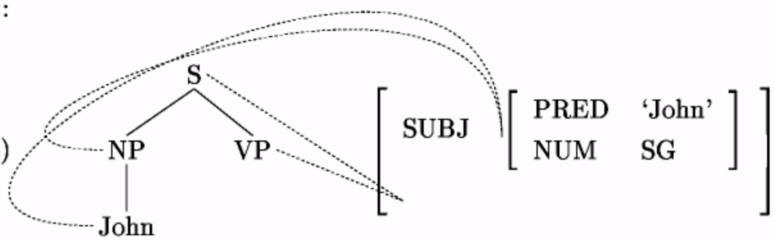
Frontier operates as in Tree-DOP, deleting all subtrees of the selected frontier nodes. Furthermore, it deletes all ϕ-links of these deleted nodes together with any semantic form corresponding to the same nodes, as in (13):

(13)



This illustrates the ability of *Root* nodes to access certain features (TENSE, here) even after subnodes have been deleted. While this may look odd to speakers of English, Bod and Kaplan (*op cit.*) note that Subject-tense agreement *is* seen in some languages (such as Hindi). Consequently, there is no universal principle which rules out fragments such as (13). Nevertheless, the above example can be pruned still further, as in (14):

(14)



by applying a third, and new operation, *Discard*, to the TENSE feature in (13). This represents directly the probability that there is no Subject-tense dependency in English, and consequently we would expect to see such fragments more frequently in our treebanks. It is this *Discard* operation that adds considerably to LFG's robustness. *Discard* provides generalised fragments from those derived via *Root* and *Frontier* by freely deleting any combination of attribute-value pairs from an f-structure except those that are linked to some remaining c-structure node via the ϕ mapping, or that are governed by the local predicate. Its introduction also necessitates a new definition of the grammaticality of a sentence ***with respect to a corpus***, namely any sentence having at least one derivation whose fragments are produced only by *Root* and *Frontier* and not by *Discard*.

Composition is also a two-step operation. C-structures are combined by leftmost substitution, as in Tree-DOP, subject to the matching of their nodes. F-structures corresponding to these nodes are then recursively unified, and the resulting f-structures are subjected to the grammaticality checks of LFG. Finally, $P(f \mid CS)$ denotes the probability of choosing a fragment f from a competition set CS of competing fragments. The probability of an LFG-DOP derivation is the same as in Tree-DOP (cf. (6) above). In Tree-DOP, apart from the *Root* and *Frontier* operations, there are no other well-formedness checks. LFG, however, has a number of grammaticality conditions. Given this, we can define probabilities only for valid representations by sampling *post hoc* only from such representations. The probability of a valid representation is (15):

$$(15) \qquad P(R \mid R\ is\ valid) = \frac{P(R)}{\sum_{R'\ is\ valid} P(R')}$$

If we choose to enforce the LFG grammaticality checks at various stages in the process, we produce a number of different competition sets. Bod & Kaplan (*op cit.*) describe four such sets linked to the **probability models** for LFG-DOP. In brief, these are:

1. A straightforward extension of the Tree-DOP probability model, where the choice of a fragment depends only on its *Root* node and not on the Uniqueness, Completeness or Coherence conditions of LFG. Consequently, unless a large number of valid representations are sampled with high conditional probabilities, this model can be expected to produce many invalid representations.

2. C-structure nodes must match, and f-structures must be unifiable if two LFG fragments are to be combined. This model takes the LFG Uniqueness condition (namely that each attribute has only one value) as well as the *Root* category into account. As the resultant fragments produced vary depending on the derivation followed, unifiability must be determined at each step in the process.

3. In addition to the steps outlined thus far, the LFG Coherence check is enforced at each step, ensuring that each grammatical function (SUBJ, OBJ etc.) present in the f-structure is governed (i.e. required to be present) by a PRED. This means we are dealing only with well-formed c-structures which correspond to coherent and consistent (i.e. which satisfy LFG's Uniqueness check, thereby permitting unification only where exactly appropriate) f-structures.

4. The final model is one where all checks—all LFG grammaticality conditions, as well as the DOP category-matching stipulation—are left to the end.

Bod & Kaplan (*op cit.*) note that in models 1-3 the category matching condition is enforced on-line, and all LFG checks are either performed on-line or *post hoc*, whereas given the non-monotonic nature of the Completeness check (i.e. that each grammatical function governed by a PRED is present in the f-structure), this can only ever be enforced *post hoc*. It is easy to envisage a number of other models where various combinations of these conditions are evaluated at different stages in the process.

3 Applications of DOP in Translation

DOP-based models can be used on several levels in the translation process:

- as an EBMT system;

- as a fully fledged MT system using DOP.

We shall discuss the merits of each of these prior to our LFG-DOP MT proposals.

3.1 DOP as an Exemplar of EBMT

One obvious yet novel use for DOP as a model for MT is as an example of EBMT, with corpora of linked source and target trees and subtrees. Such a model would act exactly like any other EBMT system, where new strings are analysed by looking up 'similar' examples of previously encountered aligned source and target sentences in a corpus, and suggestions are made as to 'fuzzy' matches which exceed some predetermined threshold for the user's selection for post-editing (if necessary) into the final 'correct' translation. DOP-based corpora would facilitate the alignment process as the syntactic information available would enable the system to better establish links between source and target chunks, rather than, as is the case with unparsed (or even untagged) corpora, leaving the system to try to determine translational equivalents in isolation of such linguistic clues.

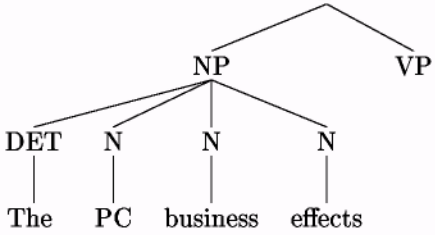
Like all statistical systems, one of the major attractions of EBMT is that the larger the corpus becomes, the greater the chance of finding a good translation match. However, when there are a large number of overlapping fragments on which to base a candidate translation, EBMT systems can be led astray by ostensibly conflicting choices; DOP can suffer in the same way, as in (17) below, but we can be relatively confident in assuming that–in the general case–correct translations are obtained given that the immediate syntactic context is always available to try to help ensure the optimal selection of ambiguous words. The two problems for EBMT in this regard are *boundary definition*—retrieved fragments may not be well-formed constituents—and *boundary friction*—the retrieval process does not take context into account, neither of which are as problematic for DOP. The former is a particular problem for 'pure' EBMT systems, in that we need to ensure syntactic well-formedness without actually employing grammatical information, but this is always available in DOP. Nevertheless, this does not always help. To give an example, in attempting to translate the sentence *The PC business effects changes in its marketing strategy for its European operations*, an EBMT system may have the fragments in (16) in its memory on which to base its judgements:

(16) a. The PC business effects are wide-ranging in the Asian economy.

b. British Telecom changes radically in its marketing strategy for the next century.

This might cause it to produce the incorrect translation with *changes* as the verb and *effects* as a noun, rather than the other way round, assuming no other suitable chunks are available. DOP too errs here: it is possible to insert the VP headed by *changes* into the tree (17):

(17)



to end up with a wrong (yet valid with respect to the corpus) analysis for this sentence. Depending on the corpus, however, it is reasonable to expect that the probability of other correct derivations may exceed that of this incorrect analysis. This shortcoming would be overcome by augmenting a DOP treebank with LFG f-structures, as LFG's unification element would prevent such a derivation with a plural SUBJ NP and a singular VP.

In such circumstances, it is generally the case that the more context is available, the better DOP is at resolving such ambiguity, as, like Probabilistic Context-Free Grammars (PCFGs), but unlike n-gram approaches, it prefers larger chunks. A PCFG would be much more likely to assign a higher probability to the string *John baked a cake* than it would the string *John baked a*, given the preference for a rule such as VP → V DET N over one like VP → V DET, whereas we can expect the opposite result to be achieved with a Hidden Markov Model, given the implausibility that $P(cake \mid John\ baked\ a)$ would have a probability of 1 in a corpus. Most, if not all, EBMT systems give extra weight to larger chunks, all else being equal, when it comes to constructing new translations (e.g. Sato & Nagao 1990; Veale & Way 1997). It is, therefore, in the same way unsurprising that DOP's accuracy in parsing increases when larger chunks are taken into account (Bod 1993).

In any case, the general criticism of length-based approaches, whether we are using an n-gram approach (e.g. Brown *et al.* 1990), or a window-based approach (e.g. Grefenstette 1993; Brown *et al.* 1992b; Gale *et al.* 1992), is the restriction imposed by the amount of context which can be handled at any one time. If we are attempting to deal with collocations, or long-distance dependencies or trying to establish the best chunk for mapping in translation, tree-based approaches such as DOP handle such phenomena within *structures*. Such dependencies have nothing to do with length, or distance, which are arbitrary, wrong notions, nor complex notions like mutual information.

3.2 Data-Oriented Translation (DOT)

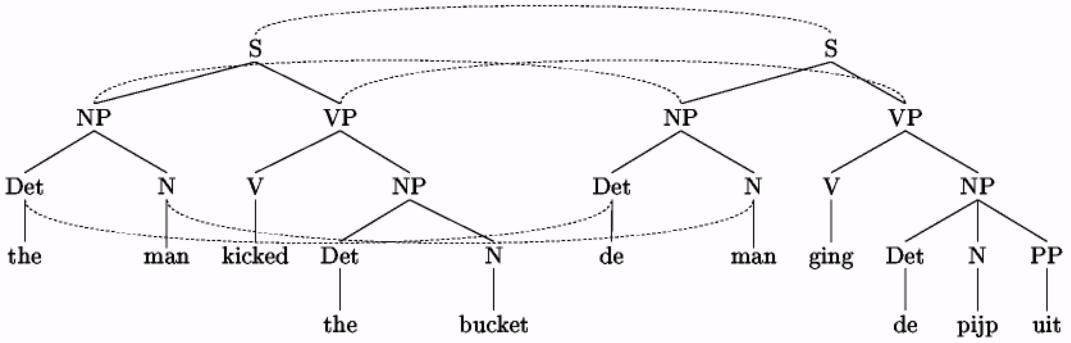
Notwithstanding the improved production of correspondences between source and target chunks that can be expected by linguistically enhanced corpora such as those provided by DOP, the development of such a static system can be viewed as an underachievement of what might be possible if we were to use the full machinery afforded by DOP. All else being equal, we would prefer a dynamic MT system, and in this spirit Poutsma (1998) has developed such a model—Data-Oriented Translation (DOT).

His DOT model relates POS-fragments between two languages (English and Dutch here), with an accompanying probability. Once a derivation for the source language sentence has been arrived at (using the methodology of DOP outlined in §2.1), the target structure is assembled, and a string produced. Since there are typically many different derivations for the source sentence, there may be as many different translations available. As is the case when DOP is used monolingually, the probability of a translation is calculated by summing the probabilities of all possible derivations forming the translation (cf. (6) above). Poutsma shows that the most probable translation can be computed using Monte-Carlo disambiguation, and exemplifies this using sentence idioms.

Poutsma's system is premised on both the Principle of Compositionality of Translation, namely that two strings are considered to be translations if and only if they have been constructed from parts which are each others translations, as well as the Principle of Compositionality of Meaning, which states that the meaning of an expression is a function of its constituent parts and the way they are combined.

That is, given the two linked trees in (18):

(18)



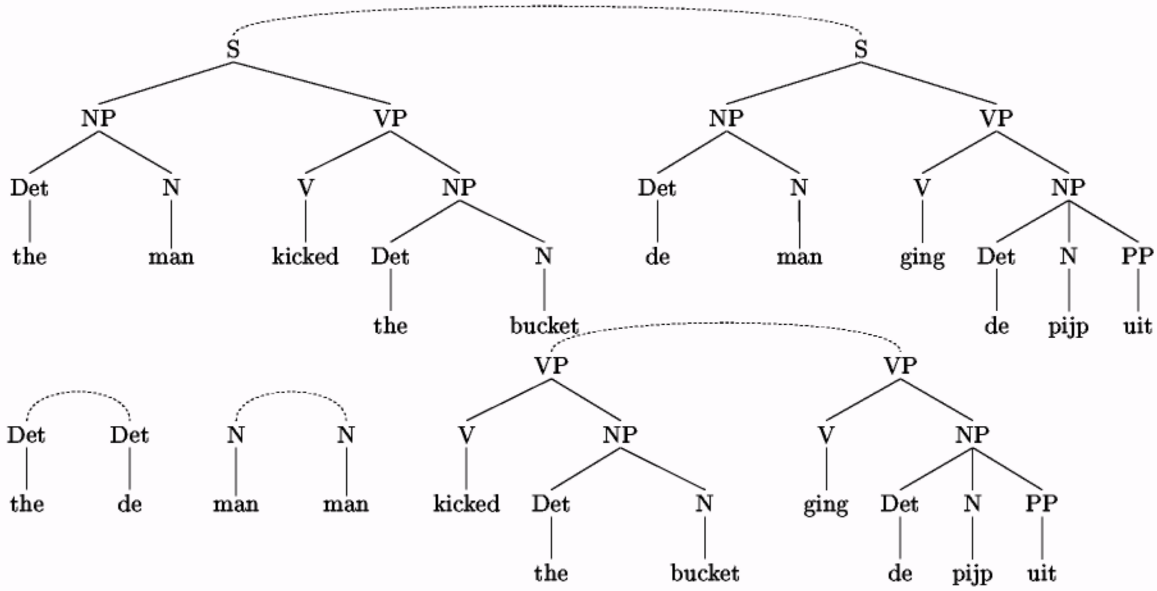
each link symbolises a semantic equivalence: any tree can be replaced with its linked translation with no loss of meaning. A link must exist at the root level, and links may exist at all levels other than at the leaves.

In order to form sub-analyses which can be used in translation, Poutsma defines how source-target DOP fragments are to be linked:

> A subtree-pair of a linked pair of trees T_1 and T_2 consists of two connected and linked subgraphs U_1 and U_2 such that:
>
> 1. for every pair of nodes U_1, U_2 for which a link exists between their corresponding nodes in T_1, T_2, it holds that:
>   (a) both U_1 and U_2 have zero daughter nodes,
>   or
>   (b) both U_1 and U_2 have all the daughter nodes of their corresponding nodes in T_1 and T_2
>   and
> 2. every node in U_1 (or U_2) that does not have a link in T_1 (or T_2) has all the daughter nodes of the corresponding node,
>   and
> 3. both U_1 and U_2 consist of more than one node,
>   and
> 4. at least one (top level) link between U_1 and U_2 must exist.

Some subtree-pairs arising from the linked trees in (18) include those in (19):

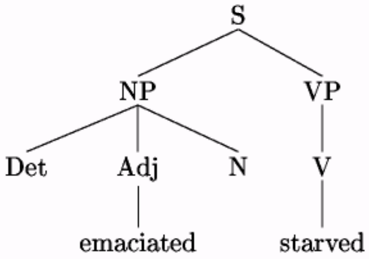
(19)



3.3 Translations in Context

For the most part, of course, the fragments produced in DOP corpora correspond exactly to PS-rules. However, this need not be the case, as in the example *the emaciated man starved*, where one of the fragments obtained would be (20):

(20)



This is one major difference between DOP and PCFGs. Given that PCFGs are necessarily based on a linguistic theory, they are restricted by the content of the rules of that theory—it is hard to conceive of a linguist writing a rule such as *S ⟶Det emaciated N starved*—so the relationship shown in (20) would not be captured. DOP has no such locality restriction, in principle at any rate (although we note that the number of resultant fragments is very large), enabling collocations to be captured which naturally occur outside of PS-rules, as here. This facility gives DOP an advantage over true rule-based systems—for instance, the *Eurotra* formalisms CAT (e.g. Arnold & des Tombe, 1987) and EF (e.g. Bech & Nygaard, 1988)—as we can provide parts of translations in context. However, in linking problematic parts of source-target fragments, DOP models *do* correspond to translation rules (which are—normally—different animals entirely from PS-rules). So we should be able to deal with 'hard' cases, and combinations of exceptions (Way *et al.* 1997), assuming, as always, that instances of the specific translations do appear in the corpus. Consider the examples in (21):

(21) a. Le gouvernement sait le faire ⇔ The government knows how to do it.

b. On le fait maintenant ⇔ It is done now.

c. On sait le faire ⇔ How to do it is known.

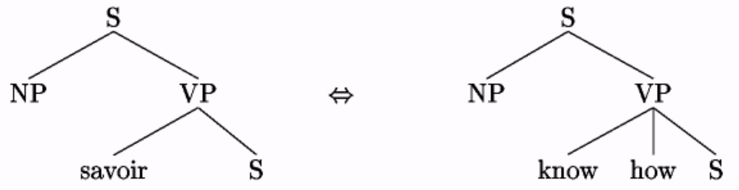
d. On mange bien en France ⇔ One eats well in France.

The point is that it is generally the case that French sentences with *on* as subject are best translated in English as agentless passives, despite examples such as (d). In brief, Way *et al.* (*op cit.*) show that the mainstream *Eurotra* formalisms cannot handle sentences like (c), which contain two problematic translation phenomena—the insertion of *how*, plus the translation of *on*—using the specific rules written for the translation of these phenomena in isolation (examples (a) and (b)), but only by writing a rule to handle the combination of such exceptions. However, Mimo (Arnold *et al.*, 1988; Arnold & Sadler, 1990) can handle such combinations of exceptions in a compositional fashion. For instance, a Mimo rule to translate cases with *savoir* would be as in (22):

(22) !savoir.[a!arg2] ⇔ !know.[!mod=how.[a!arg2]]

where only those parts which are relevant to the translation problem are mentioned. CAT (or EF) suffers in examples like these by its local tree restriction, whereby if a rule mentions any two nodes in a local tree, it must mention the whole local tree. In its corresponding rule, therefore, it is forced to mention *all* daughters of *savoir*, despite the fact that the subject position has nothing to do with the problem of translating *savoir* itself, and the presence of this node in the *savoir* rule prevents the *on* rule from firing, necessitating a new rule which includes both phenomena in the same context. This effect is pervasive and causes systems like CAT to approximate to sentence dictionaries when faced with combinations of difficult translation problems. DOT, however, is like Mimo here in that its models correctly prefer the specific translations for sentences like (21)(c) over those incorrect ones produced by default; one can see quite straightforwardly that a tree fragment (generalised in (23) with infinitives) will be produced which corresponds exactly to the Mimo rule (22):

(23)

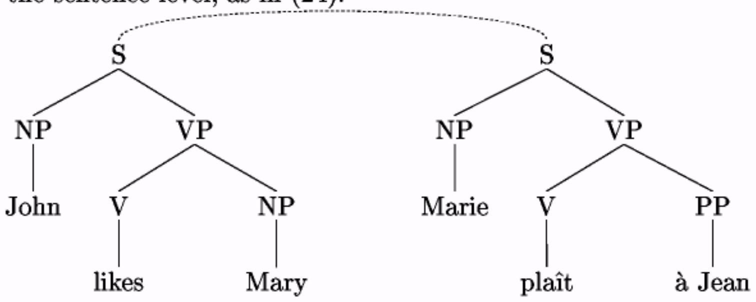


3.4 Some Limitations of DOT

It may be hoped that we might be able to safely rely on the probability model of DOT to prefer translations using specific ‘rules’ such as these to those derived via smaller fragments of default translations, as once these are combined into a complete tree, the effect of multiplication of already small probabilities renders the likelihood of these translations very low indeed compared to those derived using larger tree fragments, i.e. the words in context.

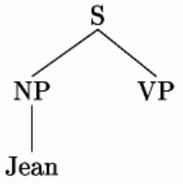
However, DOT may not produce the correct translation when faced with certain example sentences, yet when LFG-DOP MT is used instead this problem can be overcome. For example, it is unclear whether the *like* ⇔ *plaire* (relation changing) case, let alone others of greater translation difficulty, could be dealt with in such a model. That is, how can we map fragments of POS trees between source and target? Even if we can align at the sentence level, as in (24):

(24)



how can we show the links between $\langle John, Jean \rangle$ and $\langle Mary, Marie \rangle$ here? Note that interestingly, unless there is some prior occurrence of *Jean* as object, or *Marie* as subject, DOT may actually prefer the wrong translation *Jean plaît à Marie.* If we have a treebank built up from *Jean embrasse Marie* and *Sarah plaît à Bill*, then the string *Jean plaît à Marie* is about 1.25 times more likely than the correct alternative given the French language model. The reason for this is DOP's preference for *Jean* as subject, given the following tree (25) already in its treebank:

(25)



Furthermore, DOP's statistical model also gives a 'level of correctness' figure to alternative translations. This is useful (though must be treated with caution, as it may rank wrong translations above correct alternatives) in cases where the default translation in LFG-MT (and in many other systems) cannot be suppressed when the specific translation is required. For example, assuming the basic default rules in (26):

(26) a. commettre ⇔ commit

b. suicide ⇔ suicide

in order to deal with the sentences in (27):

(27) a. Jean commet un crime ⇔ Jean commits a crime

b. Le suicide est tragique ⇔ Suicide is tragic

we would get the wrong translation in (28):

(28) John commits suicide ⇔ *John commet le suicide (cf. John se suicide)

Since we require specific rules to override the default translation where applicable, in LFG-MT we would get both translations here, i.e. a correct one and a wrong one. Assuming a DOP treebank built from the French sentences in (27) as well as *Marie se suicide*, the ill-formed string *Jean commet le suicide* is preferred (in the French language model) about about half as much again as the correct alternative *Jean se suicide.*

There are several reasons for this: the preference for *Jean* as subject of *commettre*, the co-occurrence of *le* and *suicide*, plus the fact that *commettre* is followed by an NP consisting of a Det + N sequence. Note that producing more than one translation for a string is not possible with LTAG-MT (Abeillé *et al.*, 1990), for instance, so in this case we assume that the more likely French string will be proposed as the wrong, final translation.

Note also that these results are obtained with the same number of instances of each verb — in a larger corpus *commettre* would surely greatly outnumber instances of *se suicider*. Nor are they by any means unexpected. As an example, in the LOB Corpus [5], there are 66 instances of *commit* (including its morphological variants), only 4 of which have *suicide* as its object, out of the 15 occurrences of *suicide* as an NP. Consequently, even for this small sample, we can see that 94% of these examples need to be translated compositionally (by *commettre* + NP), while only the *commit suicide* examples require a specific rule to apply (i.e. *se suicider*). In the on-line Canadian Hansards [6] covering 1986-1993, there are just 106 instances of *se suicider* (including its morphological variants). There will, of course, be many thousands of instances of *commettre*. Given occurrences of *suicide* as an NP in French corpora, it is not an unreasonable hypothesis to expect that the wrong translations such as (28) will be much more probable than those derived via the specific rule.

Of course, these are by no means isolated cases of difficult translation problems. Way *et al.* (1997) produce a categorization of a number of 'hard' cases of translation containing complex insertions and deletions, such as:

1. 'Schimmel' cases (also classified as cases of 'Conflational Divergence' (Dorr 1993, p.258f), or '1-to-N lexical gaps' (Lindop & Tsujii 1991), from the well-known example where the German noun *Schimmel* translates as the English governor *horse* plus a complete AP modifier containing *white*.

2. Relation changing verbs, such as (24), *like* ⇔ *plaire*.

3. 'Shoehorn' cases, such as (21), *savoir* ⇔ *know how*, where an additional piece of structure needs to be 'shoehorned in' by the target grammars around an already existing piece of target structure.

4. Headswitching cases, such as (9), or *Ich arbeite gerne* ⇔ *I like working* (cf. note 1), where what in English is realised as a main verb is expressed in German by means of an adverbial modifier, *gerne*.

These category mismatches, lexical holes, insertions, and deletions can be described in similar terms to those used by IBM-MT (Brown *et al.*, 1990; 1992a) — *fertility* and *distortion*. Whichever description one chooses, however, the mechanics of DOT remain the same whatever pair of languages one is translating between. Nevertheless, we expect it to do better between languages with similar word-order, in a similar way to transfer, which prefers 'like' languages, whereas the interlingual approach is often quoted as being better for dissimilar ones.

However, it would appear that the adherence to left-most substitution in the target given *a priori* left-most substitution in the source is too strictly linked to linear order of words, so that, as soon as this deviates to any significant extent even between similar languages, DOT has a huge bias in favour of the incorrect translation. Even if the correct, non-compositional translation is achievable in such circumstances via DOT, it is likely to be so outranked by other wrong alternatives that it will be dismissed, unless all possible translations are maintained for later scrutiny by the user.

One line of investigation which we now develop that can overcome this linear restriction is to use LFG's τ-equations to relate translation fragments between languages.

4 LFG-DOP MT

The DOT model cannot always explicitly relate parts of the source language structure to the corresponding, correct parts in the target structure. We now propose the use of LFG-DOP (Bod & Kaplan, 1998) as the basis for an innovative MT system. LFG's τ-equations are able to link exactly those source-target elements which are translations of each other. In turn, DOP adds robustness to LFG-MT, both with respect to dealing with ill-formed input, and to dealing with well-formed input not covered by the treebank. Of course, they ought to work well in tandem here, as even if we have partial, uncompletable fragments (with respect to the corpus), the unification element in LFG may bring extra information to bear in constraining the missing element.

4.1 Two Models for LFG-DOP MT

There seem to be at least two possibilities as to how LFG-DOP MT might work.

4.1.1 Model 1: $\langle c, \phi, f, \tau, f', \phi', c' \rangle$

This is a simple, linear model. Given separate language corpora, the model builds a target f-structure f' from a source c-structure c and f-structure f, the mapping between them ϕ, and the tau-equations τ. From this target f-structure f', a target string will be generated via the standard LFG generation algorithms (cf. Wedekind 1988; Kohl 1992), as illustrated by the mapping ϕ' in (29), explicitly linking f' and c':

(29)

$$\begin{tikzcd} \mathrm{c} \arrow[r, dash, "\phi"] & \mathrm{f} \arrow[d, dash, "\tau"] \\ c' \arrow[r, dash, "\phi'"'] & f' \end{tikzcd}$$

The different components needed then are:

- a source language LFG-DOP model;
- the τ mapping;
- a target language LFG-DOP model.

Note that ϕ' is not a function: one only has to think of free word order languages to see immediately that one f-structure can represent many different strings. Given this, we intend to arrive at the most probable c-structure, and string, via regular DOP probabilistic techniques, such as Monte-Carlo sampling. There is no reason to suggest that such techniques which work well for DOP will not carry over successfully to LFG-DOP.

The advantage of this model over DOT is the availability of the explicit τ-equations to link source-target correspondences. For instance, the LFG-MT solution to the *like* ⇔ *plaire* case, (24), is (30):

(30) *like*:
$(\tau\!\uparrow \text{PRED }) = \text{plaire}$
$\tau(\uparrow \text{SUBJ}) = (\tau\!\uparrow \text{OBL})$
$\tau(\uparrow \text{OBJ}) = (\tau\!\uparrow \text{SUBJ})$

That is, the subject of *like* is translated as the oblique argument of *plaire*, while the object of *like* is translated as the subject of *plaire*.

The solution to the *commit suicide* ⇔ *se suicider* problem, (26-28), is (31):

(31) *commit*:
$(\tau\!\uparrow \text{PRED }) = \text{se suicider}$
$\tau(\uparrow \text{SUBJ}) = (\tau\!\uparrow \text{SUBJ})$
$(\uparrow \text{OBJ PRED}) =_c \text{suicide}$

Here the collocational units '*commit + suicide*' are linked as a whole to *se suicider*. The $=_c$ equation is a constraining equation: rather than expressing mere equality, it constrains the PRED value of the OBJ of *commit* to *suicide* when it is to be translated as a whole into *se suicider*.

Using LFG τ-equations ensures the derivation of the correct target f-structure, along with some wrong alternatives via the default rules. We cannot be sure that the generation of a target string via the correct target f-structure will be a more probable translation than any wrong alternative, but it will exist as one of a small number of high-ranking candidate solutions from which the final translation can be selected. Of course, the better the target language LFG-DOP model, the more likely the correct translation is to be the most preferred translation.

4.1.2 Model 2: $\langle c, f, \phi\rangle \longrightarrow \gamma, \tau \longleftarrow \langle c', f', \phi'\rangle$

Here we have integrated language corpora, where for each node in a tree c, we relate it both to its corresponding f-structure fragment f and its corresponding target c-structure node c', and for each source f-structure fragment, we relate that to its target language fragment in f-structure f', via τ, as in (32):

(32)
$$\begin{tikzcd}
\text{c} \arrow[r, dash, "\phi"] \arrow[d, dash, "\gamma"'] & \text{f} \arrow[d, dash, "\tau"] \\
c' \arrow[r, dash, "\phi'"'] & f'
\end{tikzcd}$$

Model 2 contains explicit links between both surface constituents and f-structure units in both languages, whereas Model 1 relates the languages just at the level of f-structure (via τ). Here γ is the DOT model outlined in §3.2. Consequently, Model 2 is a good deal more complex, necessitating:

- a source language LFG-DOP model;
- the γ mapping (i.e. the DOT model);
- a target language LFG-DOP model.
- a methodology whereby the γ-probabilities can be integrated with the τ mapping to derive a translation via the most probable combination of linked LFG-DOP fragments.

The principal reason for hypothesising the γ function is that it is reasonable to assume that, as Poutsma (1998) has shown, that valuable information concerning the final formulation of the target string can be influenced by the source c-structure. In this way we have two pieces of information at hand with which to build the target string—the γ and τ' functions, which if they can be properly harnessed, should bring about a better translation, given the extra evidence that has been brought to bear in its generation.

The main problem in both models is that the target f-structure built may not be acceptable to the target language model, either because it is missing as an exemplar from that model (i.e. it is unable to be deconstructed via ϕ' into a c-structure), or because given the target model, it is provably ill-formed. In these circumstances it may well be that the two corpus bags (ill-formed and well-formed) can eke out a (partial) solution, and again, given Model 2 we have the built-in DOT model on which to fall back should the ϕ' mapping fail.

Finally, either model presented here could be extended to cope with LFG σ-structures (cf. Butt 1994), with the further addition of a mapping σ' to relate target f- and σ-structures (i.e. f' and σ'), to bring still more information to bear to the translation process. This would tie in nicely with the extensions to DOP to cope with semantics which have already been developed (Bonnema *et al.* 1997), resulting in more complex models.

4.2 The *Discard* Function in LFG-DOP

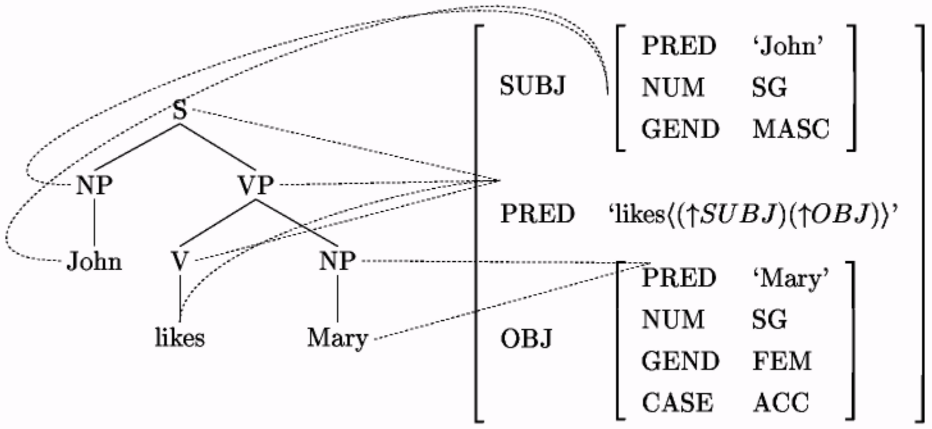
Our proposed models for MT using LFG-DOP centre on the use of the *Discard* function [7]. There are two possible interpretations of how *Discard* affects the treebank: the first possibility (P1) assumes the machinery of LFG-DOP in its entirety as outlined by Bod & Kaplan (1998), whereas the second (P2), whilst maintaining the *Discard* operation, employs it solely at what might be termed 'parse-time'.

In P1, the treebank contains all possible fragments derivable via *Root*, *Frontier*, and *Discard*. That is, the fragments derivable via *Discard* are created prior to combination, which follows on in a separate procedure. The immediate criticism which arises in such circumstances is that the treebank is potentially huge, and perhaps unmanageable. *Discard* is used solely to improve robustness, just in case we encounter ill-formed strings, or well-formed strings whose interpretations can only be derived via the generalization of f-structure fragments. However, note also that the adoption of this model will also affect combinations where there is no ill-formed input, in terms of increasing the number of potential candidate fragments participating in the combination stage, which causes the probability model to be altered. A potentially far more serious problem is the increased processing time necessary to cope with the larger bag of fragments. The good thing about maintaining such a huge treebank is that no extra routines are needed (although the search problem is an astronomical one).

Nevertheless, already for Tree-DOP (Bod 1995) the number of fragments becomes extremely large even for relatively small corpora, e.g. 10^6 fragments for the (small, at 750 sentences) ATIS [8] corpus if lexicalised fragments are used. Despite this, it was possible to *make* Tree-DOP practical, first by estimating the most likely parse via Monte-Carlo sampling (*op cit.*), then by estimating the most likely parse via the most likely derivation (Sima'an 1995, 1996). Although in this latter case the results are sub-optimal compared with those obtained 'full' processing, the parser created operates in near to real time for a corpus of 10,000 strings.

Nevertheless, if there is an explosion of fragments in DOP, then the number of fragments in LFG-DOP is potentially crippling. As Cormons (forthcoming) shows with the simple example (33):

(33)

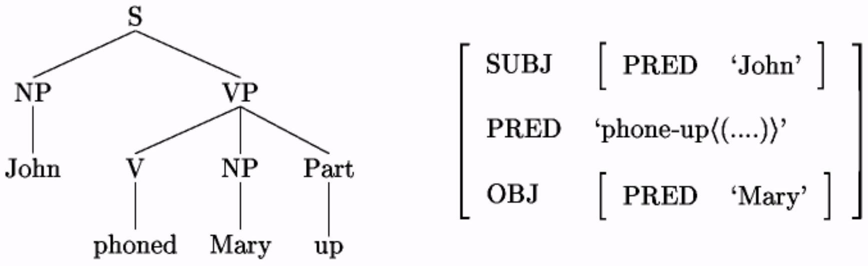


there are a potential 256 ⟨c,f,ϕ⟩ derivable fragments just for the VP, which are restricted to 32 by linking words in the c-structure with their corresponding f-structure predicates at the time of fragmentation. Notwithstanding this, it can be seen that the amounts of data we are dealing with are non-trivial.

4.2.1 Alignment Problems in LFG-DOP

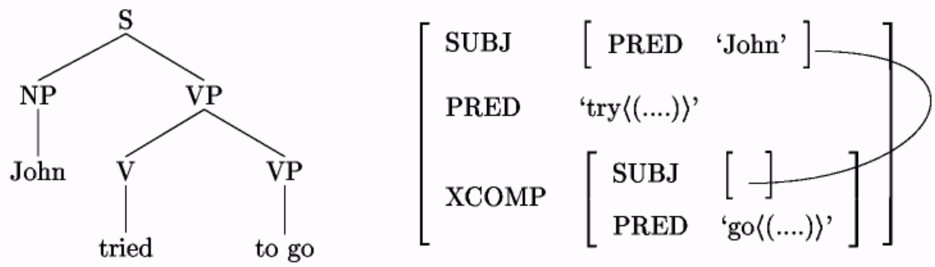
Assuming that we are able to link c-structure nodes with their corresponding f-structure attributes, there are some rather more problematic instances, for which, while being somewhat more peripheral to the argument as a whole, *some* solution needs to be found. For instance, one can have discontinuous elements in the c-structure which need to 'come together' in the f-structure, as in (34):

(34)



In addition, there are examples of elements in the f-structure which have no overt c-structure nodes to which they can be linked, such as in (35):

(35)



This latter example merits some discussion. The LFG device of functional control makes explicit the 'understood' SUBJ of *go* via an explicit equation: (↑ SUBJ) = (↑ XCOMP SUBJ), illustrating that *John* is SUBJ both of *try* and *go*. Nevertheless, we would be wrong in physically linking the *John* c-structure node to both SUBJ slots in the f-structure, as the line linking these two SUBJ slots indicates re-entrance of the structure, namely that there are two distinct paths to this structure, which are *token* (and by implication *type*) identical, which is different from having two copies of the same structure in different locations, which may be *type*, but not *necessarily token* identical.

4.2.2 Limiting the Number of Fragments

Despite the potential explosion of fragments in LFG-DOP, a number of ways of limiting this suggest themselves:

1. It is possible to distinguish between lexical features such as gender and number on the one hand, and structural features like case on the other, which could be regarded as a universal distinction. We might enforce the restriction that lexical features can only be discarded if the word (the PRED value) to which they are linked is also discarded.

2. Those tree fragments greater than depth 1 (i.e. containing some categories with no associated terminals) could be disregarded unless at least one non-terminal contains an overt lexical item as its daughter, i.e. they should be in Reibach (rather than Chomsky) Normal Form (cf. Tree Insertion Grammars—Schabes and Waters, 1995). The problem here is that in these circumstances we would most likely need to incorporate an Adjunction operation into our DOP model, as it is known that even if lexicalised CFG's can generate the same strings, they severely undergenerate with respect to structures. This would have unfortunate consequences for our models. The only difference between DOP and models like Probabilistic Tree Adjoining Grammars (PTAG—Resnik 1992) or Stochastic Tree Adjoining Grammars (STAG—Schabes 1992) is the Adjunction method of composition in TAG, which makes STAG (and PTAG) mildly context-sensitive. Given also that STAG is derived from an underlying grammar, certain word dependencies such as (20) cannot be captured by STAG, whereas DOP can handle such collocational information quite naturally. Finally, only the probability of a *derivation* can be calculated given current implementations of STAG, and we need to know also what the probability of a *tree*, and perhaps its *meaning*, is. Currently, therefore, STAG is restricted by the linguistic dependencies of the underlying grammar (a *competence* model), whereas DOP clearly favours a *performance* model.

3. We might attempt to redefine the *Discard* Function so that its effects are not so wide-ranging.

It is this latter proposal that we will now explore further.

4.3 A Reinterpretation of the *Discard* Function

Of course, we may hope that the savings in processing time for DOP carry over to LFG-DOP MT. Assuming this not to be the case, however, we advocate the adoption of P2, which strives to avoid such extra processing. *Discard* is used to derive fragments only where absolutely necessary, i.e. in those cases where generalised fragments are the only recourse to achieving a translation given the treebank derived via *Root* and *Frontier*. The treebank is, therefore, obviously much smaller than in P1. P1 is simple, but perhaps intractable due to the search problem. If we adopt P2 to try to avoid this, then the minimum criterion is that *Discard* should apply only when unification fails. This obviously entails further computation, as we need to stipulate what 'unification failure' means; for example, we want *Discard* to operate when we have a clash in value for a given attribute such as NUM=SG and NUM=PL, but not when we have a clash of PRED values, such as PRED=John and PRED=Mary. There must be a countable number of instances of such cases—one can think of subject-verb agreement, relative-clause agreement between the verb and the modified NP, as well as between this NP and the relative pronoun. Others will include all cases where there is a chain of derivation, i.e. including all movement phenomena, and so on. Once this list has been established, we envisage three ways in which composition via *Discard* could work:

1. Use *Discard* every time any such unification failure is encountered.

   The problem here is that there would be lots of redundancy, and the amount of computation is potentially crippling. The advantage is that such computation is done on the fly, so that no storage of the extra fragments derived would be needed.

2. For all structures in the treebank affected by such a unification failure, perform *Discard* on all such structures, and put the results into an 'ill-formed bag' (IFB), leaving the original treebank unaffected, i.e. nothing goes into the 'well-formed bag' (WFB)—the original treebank—which has been produced by a *Discard* operation.

   The problem with this scenario is that there is a lot of computation, but it only needs to be done once for each type of unification failure. The merit of such an approach is that as a distinction is made between what is and what is not grammatical, the separate IFB is available for subsequent ill-formed input to use, making it more probable that a 'correct match' (or at least some useful bits of structure) will be found for these, analogously to EBMT (cf. §3.1). Note also that the definition of grammaticality (cf. §2.3) is maintained here.

3. For all structures encountered subsequent to a unification failure, that would be affected in the same way, perform *Discard* on all such structures in the treebank, and calculate probabilities on these affected structures only.

   The main problem with this approach is that all structures already encountered which had unification failures need to be stored, so that other similar instances can be identified in the future. The main advantage is that the whole treebank is not affected by the *Discard* operation; only 'similarly affected' structures are subjected to the generalization process.

No matter which option is selected here, we have assumed (unrealistically) that the treebank size remains constant, i.e. there is no dynamic addition to the treebank. If this were to happen, i.e. if we add a new tree and associated f-structure (and the fragments derived via *Root* and *Frontier*) to the treebank, in theory we should be able to delete some fragments from the IFB. However, in practice this is not a problem because if a hierarchical model is posited in which the WFB is searched before the IFB, the new well-formed fragments in the WFB will obviate the need to go to the IFB at all in such cases, although there may be an element of redundancy in such a model.

One more urgent problem is the definition of the circumstances whereby the IFB is to be visited. We propose that once the WFB has been left, it is impossible to return until the analysis of the next sentence. This restriction is necessary because sentences can contain more than one unification error, so that if one tries to resolve the first error encountered by going to the IFB, when trying to resolve subsequent errors, one would need to go back to the WFB, thereby potentially bringing back fragments derived from the IFB into the WFB, and we want to keep this pristine. The stipulation is, therefore, that one only goes to the IFB when a fragment with *Root=s* has been fully evaluated in the WFB.

These are all ideas for future investigation. It is clear that the *Discard* function of Bod & Kaplan (1998) is far too unconstrained, so that its application needs to be limited in some way to control the forseen explosion of fragments. The suggested solutions, at least one of which needs to be incorporated into our final LFG-DOP translation model, can perhaps only be best evaluated by implementation. We intend to report fully on the results of our investigations when these become available.

4.3.1 The Role of *Discard* in each stage of the Translation Process

Finally here, let us convince ourselves of the contribution of the *Discard* function to the robustness issue. In traditional MT systems, three main distinctions are made between different phases of the translation process—parsing, transfer, and generation. How does *Discard* get us robustness in each of these phases?

The first of these needs little further discussion, as this is the primary reason for its importation in the first place: namely, parsing ill-formed input, and dealing with unknown words and structures. Of course, to the system these are one and the same. So, with respect to MT, in dealing with the source language the task of the syntactic disambiguation component is the calculation of the probability distribution of the various probable parses. Bod (1995) outlines possible methods of dealing with unknown words, where Tree-DOP removes terminals from trees to at least allow POS categories to be estimated. This, of course, helps provide *a* (hopefully correct) context for DOP to allow further processing to continue. However, Bod (1995) notes that the biggest problem is not the parsing of unknown words, but rather the processing of items which are contained in the corpus, but for which other grammatical categories are required.

In the transfer phase (that is, where the two languages interface with one another in the system), the effect of *Discard* in parsing has knock-on effects here too. If an interpretation for some unknown element has been correctly assigned (i.e. surpasses some user-defined threshold, say), then this should help ensure the correct selection of the lexical item on the target side. Given also that all source fragments will be aligned with their target counterparts, *some* translation will always be found, if all source words are contained in the corpus. This may not, however, be the correct translation, owing to the possibility of other translations for ambiguous words not (yet) covered in the corpus, or even lexical gaps. In all such situations, of course, rule-based methods have nothing to say.

When it comes to generating the target string, the robustness brought to bear by *Discard* in parsing the source text may also help ensure the correct translation on the target side. Given that our LFG-DOP translation models are envisaged as reversible, *Discard* contributes both on the source and target sides in a similar manner.

4.3.2 A Summary of the two Interpretations of the *Discard* Function

The adoption of P1 (i.e. using Bod & Kaplan's formulation of *Discard*) has a major problem when it comes to searching the treebank because of the explosion of fragments, which we note is particularly a problem for LFG-DOP. We therefore propose to adopt (some variant of) P2. However, there are many questions to be resolved before doing so. Nevertheless, if we adopt P2, the benefits are:

1. The problem of explosion of fragments we get via P1 is alleviated considerably, i.e. the treebank (WFB) stays the same size. In addition, P1 adversely affects the processing of well-formed input, which P2 avoids.

2. Depending on the number of types of ill-formed input (i.e. the small number which cause unification failure), the IFB would not be too big, so the added complexity of the hierarchical model introduced here does not seem to be onerous compared to the vast reduction in the number of fragments assumed in P1.

3. Just as with EBMT systems, the larger the IFB, the greater the chance of finding a good match for the ill-formed input one might be confronted with in the future, which cuts down the amount of processing required by the *Discard* operation as the size of the treebank increases.

4.4 Semi-automatic Generation of DOP & LFG-DOP Corpora

Most of the examples presented here are illustrated by means of very small corpora, but nevertheless even at this level they demonstrate issues which lead us to favour LFG-DOP MT over DOT, for example. A major problem for researchers interested in LFG-DOP is the absence of suitable, extensive corpora.

Given this, in order to demonstrate further the feasibility of LFG-DOP MT, we have begun to develop our own DOP and LFG-DOP corpora (Van Genabith & Way, forthcoming). Initially we took the publicly available set of 100 sentences of the parsed AP corpus (cf. note 5.). Despite its small size, this was sufficiently large to demonstrate the plausibility of our approach. An example of such a sentence is (36):

(36)
```
A001  39 v
[N The_AT march_NN1 N][V was_VBDZ [J peaceful_JJ J]V] ._.
```

We then automatically extract the rules from this corpus (after automatically pre-processing some of the input to make it Prolog compatible), create automatically LFG-macros for each lexical category, annotate the extracted rules with LFG functional schemata by hand, and reparse the original set of sentences (recast in Prolog). For the sentence in question, *the march was peaceful*, the relevant annotated rules and lexical items are as in (37):

(37)
```
lex(at(the)).
lex(nn1(march)).
lex(vbdz(was)).
lex(jj(peaceful)).

rule(n(A), [at(B),nn1(C)]) :-
   A === B,
   A === C.

rule(j(A), [jj(B)]) :-
   A === B.

rule(v(A), [vbdz(B),j(C)]) :-
   A === B,
   A:vcomp === C.

rule(sent(A), [n(B),v(C)]) :-
   A:subj === B,
   A      === C.
```

We are thus able to produce source f-structures, as in (38):

(38)
```
sent(n(at(the),nn1(march)),v(vbdz(was),j(jj(peaceful))))

subj : spec : the
       pred : march
       num : sg
vcomp : pred : peaceful
tense : past
pred : be
```

Given the complexity of some of the strings and accompanying structures, producing f-structures automatically in this manner may be easier than producing them by hand on the fly, for a corpus of any real size. Of course, anyone working in corpus-based techniques remains open to the vagaries of the coding of that corpus, so that misparses cause incorrect f-structures to be produced, but very few other errors are introduced by our automatic procedure.

In order to produce target f-structures, all that is necessary is to add τ-equations to the lexical and structural rules, and reparse the input strings. For example, the German target f-structure corresponding to the input sentence in (36) is (39):

```
(39)    tau : subj : spec : die
                     pred : demonstration
                     num : sg
              vcomp : pred : ruhig
              tense : past
              pred : sein
```

Now that these exist, we intend to use one of the standard LFG generation algorithms (cf. Wedekind, 1988; Kohl, 1992) to produce target c-structures, and (via the ϕ' operation outlined above) target strings.

4.4.1 Interim Results

At this juncture we have produced a monolingual LFG-DOP corpus based on the 100 AP sentences, and are satisfied that our methodology is sound. It is also portable to other corpora, which is important given that the larger AP corpus is unavailable. We are about to switch to a publicly available corpus, either the LOB Corpus (cf. note 5), or the Penn Treebank (cf. note 3). Once this has been done, we shall publish further results. Nevertheless, even at this stage we feel that the bones of the system exist, and foresee no real impediment to producing a large, working system based on a substantial corpus incorporating fully the ideas outlined here.

5 Conclusions

None of the main exemplars of MT system have been particularly successful in solving difficult translation problems. A new MT system — DOT — is based on DOP, which has been presented as a promising paradigm for NLP. Despite provably deriving the most probable translation, DOT is not guaranteed to produce the best, or even a correct translation, since it is unable to explicitly link exactly those fragments which play the decisive role in translation.

Bod & Kaplan (1998) have shown how DOP and LFG can be integrated to provide a powerful mechanism for the treatment of parsing. We have described how such a model may be extended to provide a reversible, hierarchical solution to the problems of MT in the spirit of the current trend for hybrid approaches. LFG-DOP MT promises to improve on previous attempts at using LFG for translation, particular where robustness is concerned, being able to handle both unseen and ill-formed input with relative ease. It also ensures that the correct target f-structure is input into the generation process. It is reasonable to expect it to outperform pure statistics-based systems, having the additional facility of grammatical information at hand to use where necessary, as well as being able to provide collocational information outside the realms of systems with a more limited linguistic undercarriage.

Nevertheless, given the unmanageable number of fragments, we feel compelled to reject the *Discard* function of Bod & Kaplan (1998) as too unconstrained. We reinterpret this to cut down considerably on the number of fragments obtained. This prevents the ill effects of *Discard* when dealing with well-formed input by maintaining a distinction between well-formed and ill-formed fragments.

Much of this work is of course ongoing, and a number of issues remain for the future, in particular those of alignment of c-and f-structure fragments and the subsequent control of the explosion of hypotheses in LFG-DOP. Suggestions as to how these problems may be best addressed have been provided for DOP, yet it remains a focus of further study as to whether these techniques, as well as other possible solutions outlined here, carry over completely to LFG-DOP. Finally, of course, having found a workable solution to the absence of LFG-DOP corpora, we seek to finalise the development of the systems described, leading to greater experimentation on a larger scale.

References

[1] Abeillé, A., Y. Schabes and A. K. Joshi (1990) 'Using Lexicalised Tags for Machine Translation', in COLING: *13th International Conference on Computational Linguistics*, Helsinki, vol. 3, pp.1–6.

[2] Arnold, D. and L. des Tombe (1987) 'Basic Theory and Methodology in EUROTRA', in S Nirenburg, ed., *Machine Translation: Theoretical and Methodological Issues*, Cambridge University Press, Cambridge, pp.114–135.

[3] Arnold, D., S. Krauwer, L. des Tombe and L. Sadler (1988), ''Relaxed' Compositionality in Machine Translation', in *Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*, Carnegie Mellon University, Pittsburgh, pp.61–81.

[4] Arnold, D. and L. Sadler (1990) 'The Theoretical Basis of MiMo', *Machine Translation*, **5**:195–222.

[5] Beaven, J. (1992) 'Shake–and–Bake Machine Translation', in COLING: *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France, pp.603–609.

[6] Bech, A. and A. Nygaard (1988) 'The E–Framework: A Formalism for Natural Language Processing', in COLING: *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, pp.36–39.

[7] Bennett, W. and J. Slocum (1985): 'The LRC Machine Translation System', *Computational Linguistics* **11**:111–121.

[8] van den Berg, M., R. Bod and R. Scha (1994): 'A Corpus-Based Approach to Semantic Interpretation', in *Proceedings of 9th Amsterdam Colloquium*, Amsterdam, The Netherlands.

[9] Bod, R. (1992): 'A Computational Model of Language Performance: Data-Oriented Parsing', in COLING: *Proceedings of the 14th International Conference on Computational Linguistics*, Nantes, France, **3**:855–859.

[10] Bod, R. (1993): 'Using an Annotated Corpus as a Stochastic Grammar', in *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, The Netherlands, pp.37–44.

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

[12] Bod, R. and R. Kaplan (1998): 'A Probabilistic Corpus-Driven Model for Lexical-Functional Analysis', in COLING: *Proceedings of the 17th International Conference on Computational Linguistics and 36th Conference of the Association for Computational Linguistics*, Montreal, Canada, **1**:145–151.

[13] Bonnema, R., R. Bod and R. Scha (1997): 'A DOP Model for Semantic Interpretation', in *34th Conference of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, Madrid, Spain, pp.159–167.

[14] Brown, P., J. Cocke, S. Della Pietra, F. Jelinek, V. Della Pietra, J. Lafferty, R. Mercer and P. Rossin (1990): 'A Statistical Approach to Machine Translation', *Computational Linguistics* **16**:79–85.

[15] Brown, P., S. Della Pietra, V. Della Pietra, J. Lafferty and R. Mercer (1992a): 'Analysis, Statistical Transfer, and Synthesis in Machine Translation', in *4th International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada, pp.83–100.

[16] Brown, P., V. Della Pietra, P. DeSouza, J. Lai and R. Mercer (1992b): 'Class-based n-gram Models of Natural Language', *Computational Linguistics* **18**:467–479.

[17] Butt, M. (1994): 'Machine Translation and Complex Predicates', in *Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 94)*, Österreichische Gesellschaft für AI, pp.62–71.

[18] Carbonell, J., T. Mitamura and E.H. Nyberg 3rd (1992): 'The KANT Perspective: A Critique of Pure Transfer (and Pure Interlingua, Pure Statistics,...)', in *4th International Conference on Theoretical and Methodological Issues in Machine Translation*), Montreal, Canada, pp.225–235.

[19] Cormons, B. (forthcoming): *Analyse et Disambiguation: Une approche purement à base de corpus (Data-Oriented Parsing) pour le formalisme des Grammaires Lexicales Fonctionelles*, PhD thesis, Université de Rennes, France.

[20] Dalrymple, M., J. Lamping and V. Saraswat (1993): 'LFG Semantics via Constraints', in *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, The Netherlands, pp.97–105.

[21] Dorr, B-J. (1993): *Machine Translation: A View from the Lexicon*, MIT Press, Cambridge, Mass.

[22] Gale, W., K. Church and D. Yarowsky (1992): 'A Method for Disambiguating Word Senses in a Large Corpus', *Computers in the Humanities* **26**:415–439.

[23] Goodman, K. and S. Nirenburg, (1991): *The KBMT Project: A Case Study in Knowledge-Based Machine Translation*, Morgan Kaufman, San Mateo, California.

[24] Grefenstette, G. (1993): 'Evaluation Techniques for Automatic Semantic Extraction: Comparing Syntactic and Window-based Approaches', Technical Report, Dept. Computer Science, University of Pittsburgh.

[25] Grishman, R. and M. Kosaka, (1992): 'Combining Rationalist and Empiricist Approaches to MT', in *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada, pp. 263–274.

[26] Kaplan, R. and J. Bresnan, (1982): 'Lexical Functional Grammar: A Formal System for Grammatical for Grammatical Representation', in J. Bresnan (ed.) *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, Mass., pp.173–281.

[27] Kaplan, R., K.. Netter, J. Wedekind and A. Zaenen (1989): 'Translation by Structural Correspondences', in *Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, pp.272–281.

[28] Kaplan, R. and J. Wedekind (1993): 'Restriction and Correspondence-based Translation', in *Sixth Conference of the European Chapter of the Association for Computational Linguistics*, Utrecht, The Netherlands, pp.193–202.

[29] Kohl, D. (1992): 'Generation from Under- and Overspecified Structures', in COLING: *14th International Conference on Computational Linguistics*, Nantes, France, pp.686–692.

[30] Landsbergen, J. (1989): 'The Rosetta Project', in *Second MT Summit*, Munich, pp.82–87.

[31] Lehmann, H. and L. Ott, (1992): 'Translation Relations and the Combination of Analytical and Statistical Methods in Machine Translation', in *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada, pp.237–248.

[32] Lindop, J. and J-I. Tsujii (1991): 'Complex Transfer in MT: A Survey of Examples', CCL/UMIST Report 91/5, Centre for Computational Linguistics, UMIST, Manchester.

[33] Nirenburg, S., J. Carbonnel , M. Tomita and K. Goodman (1992): *Machine Translation: A Knowledge-Based Approach*, Morgan Kaufman, San Mateo, California.

[34] Poutsma, A. (1998): 'Data-Oriented Translation', in *Ninth Conference of Computational Linguistics In the Netherlands*, Leuven, Belgium.

[35] Rajman, M. (1995): 'Approche Probabiliste de l'Analyse Syntaxique', *Traitement Automatique des Langues* **36**.

[36] Resnik, P. (1992): 'Probabilistic Tree-Adjoining Grammar as a Framework for Statistical Natural Language Processing, in COLING: *14th International Conference on Computational Linguistics*, Nantes, France, **2**:418–424.

[37] Roche, E. and Y. Schabes (1995): 'Deterministic part-of-speech tagging with finite-state transducers', *Computational Linguistics* **21**:227–253.

[38] Rosetta, M. T. (1994): *Compositional Translation*, Kluwer, Dordrecht, The Netherlands.

[39] Sadler, L., I. Crookston, D. Arnold and A. Way (1990) 'LFG and Translation', in *Third Conference on Theoretical and Methodological Issues in MT*, University of Texas, Austin, pp.121–130.

[40] Sadler, L., I. Crookston and A. Way (1989) 'Co-description, projection, and 'difficult' translation', Working Papers in Language Processing 8, Department of Language and Linguistics, University of Essex, Colchester.

[41] Sato, S. and M. Nagao, (1990): 'Towards Memory-based Translation', in COLING: *13th International Conference on Computational Linguistics*, Helsinki, Finland, **3**:247–252.

[42] Schabes, Y. (1992): 'Stochastic Lexicalized Tree-Adjoining Grammars', in COLING: *14th International Conference on Computational Linguistics*, Nantes, France, **2**:426-432.

[43] Schabes, Y. and Waters (1995): 'Tree Insertion Grammar-A Cubic-Time, Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced', *Computational Linguistics* **21**:479–513.

[44] Sima'an, K. (1995): 'An optimized algorithm for Data-Oriented Parsing', in *First International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria.

[45] Sima'an, K. (1996): 'Computational Complexity of Probabilistic Disambiguation by means of Tree Grammars', in COLING: *16th International Conference on Computational Linguistics*, Copenhagen, Denmark.

[46] Su, K-Y. and J-S. Chang, (1992): 'Why Corpus-Based Statistics-Oriented Machine Translation', in *Fourth International Conference on Theoretical and Methodological Issues in Machine Translation*, Montreal, Canada, pp.249–262.

[47] Sumita, E., H. Iida and H. Kohyama (1990): 'Translating with examples: a new approach to Machine Translation', in *Third International Conference on Theoretical and Methodological Issues in Machine Translation*, Austin, Texas, pp.203–212.

[48] Tugwell, D. (1995): 'A State-Transition Grammar for Data-Oriented Parsing', in *Seventh European Conference on Computational Linguistics*, Dublin, Ireland, pp.272–277.

[49] Van Genabith, J., A. Frank and M. Dorna (1998): 'Transfer Constructors', in *Proceedings of LFG-98*, Brisbane, Australia, pp.190–205.

[50] Van Genabith, J. and A. Way (forthcoming): 'Semi-Automatic Generation of F-Structures from Treebanks, to appear in *Proceedings of LFG-99*, Manchester, UK.

[51] Veale, T. and A. Way (1997): 'Gaijin-A Bootstrapping Approach to Example-Based Machine Translation', in *Second International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, Bulgaria, pp.239–244.

[52] Way, A., I. Crookston and J. Shelton (1997): 'A Typology of Translation Problems for Eurotra Translation Machines', Machine Translation **12**:323–374.

[53] Wedekind, J. (1988): 'Generation as Structure Driven Derivation', in COLING: *12th International Conference on Computational Linguistics* , Budapest, Hungary, pp.732–737.

[54] Whitelock, P. (1992): 'Shake-and-bake translation', in COLING: *14th International Conference on Computational Linguistics*, Nantes, France, **2**:784–791.

Acknowledgements

Much of this work was done while at the CLMT group at the University of Essex. Thanks to Doug Arnold, Frederik Fouvry and Louisa Sadler for useful discussion there. In addition, thanks are due to Rens Bod, Ron Kaplan, Boris Cormons, Josef Van Genabith and Tony Veale for further helpful comments. Thanks are also due to the anonymous reviewers whose comments and criticism did much to improve this work. Any remaining errors or omissions are of course my fault. Finally, thanks to DCU for funding this research.

1. Compare the sentences:

- John likes to swim.
- John schwimmt gerne.

In the English sentence, we see that *like* is the main verb, with *swim* occurring in the complement clause. In German, however, we see that *schwimmen* is the main verb, with the 'liking' element portrayed by the adverb *gerne*. *Like* ⟶*gerne* can be handled interlingually—but in truly interlingual systems it must be handled **neutrally**! This could be with *like* or *schwimmen* as head: the first option would mirror the English, whereas the second would be more like the German, but neither would be neutral.

2. The question as to how ill-formed a text has to be before any processing of it is unmerited is merely noted here. Note also that it is possible (though unlikely) for such ill-formed data to skew the results obtained by such a system. The reader will note that this concern ultimately influences our proposals for models of LFG-DOP MT.

3. `http://linc.cis.upenn.edu/~treebank/home.html`

4. `http://grid.let.rug.nl:4321/`

5. `http://www.hit.uib.no/icame.html`

6. `http://www-rali.iro.umontreal.ca/TransSearch/TS-simple-uen.cgi`

7. Note that if we were to omit *Discard* from a model of MT, we would have a model analogous to EBMT, but it would differ from that posited in §3.1 above as it contains elements of f-structure in addition to PS-tree fragments, i.e. fragments of $\langle c,f,\phi\rangle$.

8. `http://www.cis.upenn.edu/ldc/ldc_catalog.html#atis`