Chapter 16

TRANSLATING WITH EXAMPLES: THE LFG-DOT MODELS OF TRANSLATION

Andy Way
away@computing.dcu.ie

Abstract Machine Translation (MT) systems based on Data-Oriented Parsing (DOP: Bod, 1998) and LFG-DOP (Bod & Kaplan, 1998) may be viewed as instances of Example-Based MT (EBMT). In both approaches, new translations are processed with respect to previously seen translations residing in the system's database. We describe the DOT models of translation (Poutsma, 1998, Poutsma, 2000) based on DOP. We demonstrate that DOT1 is not guaranteed to produce the correct translation, despite provably deriving the most probable translation. The DOT2 translation model solves most of the problems of DOT1, but suffers from limited compositionality when confronted with certain data. Notwithstanding the success of DOT2, any system based purely on trees will ultimately be found wanting as a general solution to the wide diversity of translation problems, as certain linguistic phenomena require a description at levels deeper than surface syntax. We then show how LFG-DOP can be extended to serve as a novel hybrid model for MT, LFG-DOT (Way, 2001), which promises to improve upon DOT and other EBMT systems.

Keywords: hybridity, constraint-based grammars, probabilistic language models

1. Problems for EBMT

One of the major advantages which is often claimed of EBMT is that the overall quality of translation increases incrementally as the set of stored translations increases. For example, Mima *et al.*, 1998 report that in their EBMT system, translation quality rose in an almost linear fashion, from 30% with 100 examples to 65% with all 774 examples (cf. Chapter 1, this volume). They also note that there seems to be a limit beyond which adding further examples does not improve the overall translation quality.

While the chances of finding an exact match become greater as the corpus size increases, there are two knock-on effects whose impact on the EBMT system should be minimized. Many models of example-based MT store examples as annotated linguistic structures. These include:

- Dependency Trees (e.g. Sato & Nagao, 1990; Yamamoto & Matsumoto, Chapter 13, this volume; Watanabe, 1992, Watanabe, 1994, cf. Chapter 14, this volume; Menezes & Richardson, Chapter 15, this volume);
- CFG-trees (e.g. Poutsma, 1998, Poutsma, 2000, cf. section 2 below; Watanabe & Takeda, 1998);
- Lexicalised Tree-Adjoining Grammar Trees (e.g. Abeillé *et al.*, 1990);
- HPSG structures (e.g. Schäler, 2001);
- LFG c-structure trees and f-structures (e.g. Way, 2001, Way, 2003).

Firstly, adding more examples has a computational cost in all EBMT systems, but this is especially the case if the examples need to be parsed into linguistic representations such as those outlined above. Adding more examples also causes storage problems, and adds to the complexity of the search and retrieval stages of the EBMT process. It is, therefore, unclear that one of the purported advantages of EBMT, namely a lessening in computational cost when the size of the translation database is increased, is indeed a real benefit. Secondly, adding more examples may not be useful in practice. For example, a newly added translation pair may be identical to, or overlap other examples. Where a system involves the computation of a 'similarity metric' (e.g. Somers *et al.*, 1994), this may be influenced by the frequency of examples, so that the score attached to certain matches increases if a large number of similar examples are found. Alternatively, in other systems, identical or similar examples may just be redundant. Somers (Somers, 1999:121; cf. Chapter 1, this volume) observes that "in such systems, the examples can be seen as surrogate 'rules', so that, just as in a traditional rule-based MT system, having multiple examples (rules) covering the same phenomenon leads to overgeneration".

The two main problems for EBMT are *boundary definition* and *boundary friction.* The first of these describes the scenario where retrieved

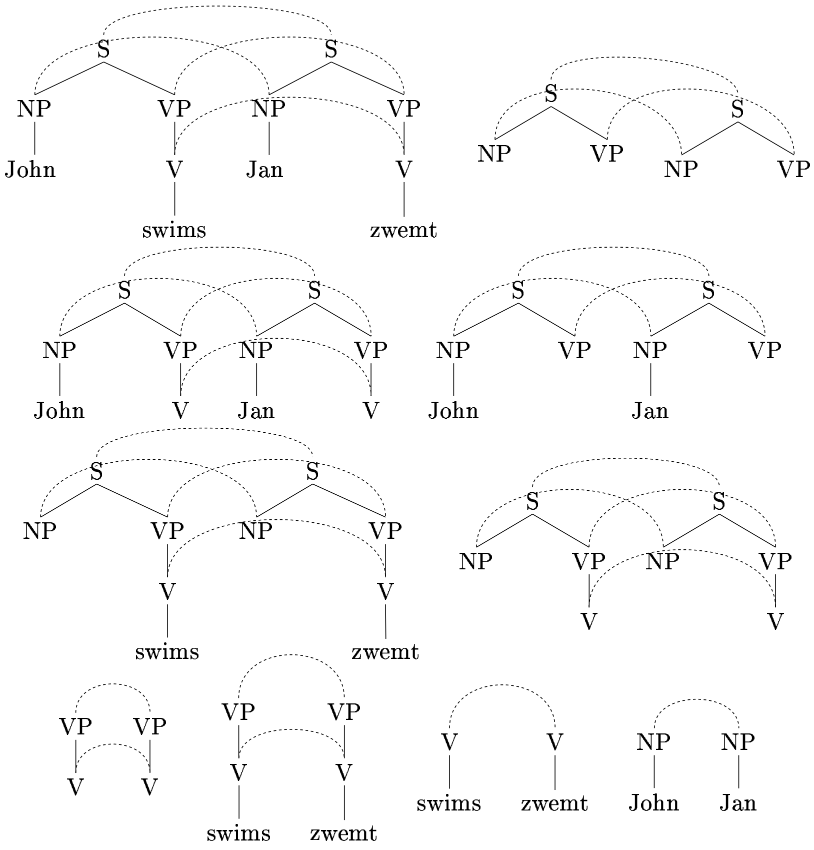


Figure 16.1. The complete DOT treebank for the linked sentence pair ⟨*John swims*, *Jan zwemt*⟩

fragments may not be well-formed constituents. This is a particular problem for 'pure' EBMT systems, where syntactic well-formedness needs to be ensured without grammatical information actually being employed. The second, boundary friction (cf. also Chapters 1 and 14, this volume), is a problem in the retrieval process, in that context may not be taken into account. This may be illustrated by attempting to translate *I have a small dog* into German. An EBMT system might retrieve the close matches in (1):

(1) a. A small dog eats a lot of meat ⟶ Ein kleiner Hund frisst viel Fleisch.

b. I have two ears ⟶ Ich habe zwei Ohren.

From these examples, the useful translation fragments in (2) might be isolated by the EBMT system:

(2) a. A small dog ⟶ Ein kleiner Hund

b. I have ⟶ Ich habe

In this case, these fragments would be combined to give the translation *Ich habe ein kleiner Hund*, which is ungrammatical as we have an NP bearing nominative case in object position.

We shall show how DOT and LFG-DOT fare with these two main problems for EBMT. With strict notions of decomposition of fragments, the problem of boundary definition is unproblematic. DOT, however, may suffer from the problem of boundary friction, while given the additional syntactic information available in the Lexical-Functional Grammar (LFG: Kaplan & Bresnan, 1982) f-structures, this problem is considerably reduced in LFG-DOT.

2. Data-Oriented Translation

Poutsma, 1998 and Poutsma, 2000 present a model of translation based on Tree-DOP—Data-Oriented Translation (DOT). There are two different versions of DOT. We shall describe these briefly.

2.1 DOT1

Poutsma's DOT1 model (Poutsma, 1998, Poutsma, 2000) is based on the methodology of Tree-DOP (e.g. Bod, 1998), and relates POS-fragments between two languages (English and Dutch here), with an accompanying probability. DOT1 is parameterized on similar lines to

Tree-DOP. Its **representations** are *linked* phrase-structure trees. Figure 16.1 shows the complete treebank for the linked sentence pair ⟨*John swims*, *Jan zwemt*⟩.[1]

These trees are augmented to incorporate semantic information, as a DOT1 treebank links semantically equivalent trees. It can be seen that the top left tree pair in Figure 16.1 represents the full parse trees for this translation pair. This tree pair is subjected to the DOT1 **decomposition** process (based on the DOP *Root* and *Frontier* fragmentation operations—essentially *Root* allows new tree fragments to be built by selecting a node to be the root node of a new tree, and deleting all other nodes except this new root and all nodes dominated by it, while *Frontier* selects a (possibly empty) set of nodes in the newly created subtree, excluding the root, and deletes all subtrees dominated by these selected nodes), so that the other tree pairs in Figure 16.1 are derived. Note also here that all S-rooted linked structures are derived via *Frontier*, while all others are produced via *Root*. It is these operations which delimit the boundary definitions in the DOT translation models.

Tree pair fragments are subjected to the **recombination** process in DOT using the same composition operation, namely leftmost substitution (in order to ensure the uniqueness of each derivation), as Tree-DOP (cf. the $\circ$ operator in (4), among other examples below). The target derivations are then assembled by replacing all subtrees of the source derivation by their linked target subtrees. Any pair of fragments which can legitimately combine with others results in a well formed derivation given the corpus. This causes a particular problem of boundary friction in DOT (cf. Figure 16.5) which is avoided in LFG-DOT (cf. Figure 16.7). Finally, the **probability** of target trees that are candidate translations is then calculated using the Tree-DOP probability model based on their relative frequencies in a treebank such as Figure 16.1. Importantly, therefore, DOT treebanks are *bags* of fragments, rather than sets. Accordingly, where duplication of 'similar' examples may be an impediment in other example-based systems, in DOT (and LFG-DOT) they are essential, as an increase in frequency of particular fragments directly contributes to the weight (in terms of probability) of the derivations in which they are involved. The probability of the parse tree of a particular translation is calculated by summing the probability of all possible derivations resulting in that parse tree. If different parse trees result for a particular translation, the respective probabil-

[1] Here, and in future examples, we 'translate' proper names purely in order to differentiate completely source and target representations and strings.

ities of each with respect to the others can be calculated with respect to the corpus. Similarly, if different translations are presented as candidate target strings, their respective probabilities are given. As will see with respect to (11)–(16) below, this is useful as certain MT systems (e.g. LFG-MT: Kaplan *et al.*, 1989; cf. section 4 below) cannot prevent the output of multiple translation candidates, even where some of these may be ungrammatical target strings. In such cases, an expert user is required to sift through these output strings to select the 'best' translation. This situation is avoidable where translations are output with probabilities, as in DOT and LFG-DOT, as translations may be ranked automatically (or pruned, if only the highest ranked translation is required).

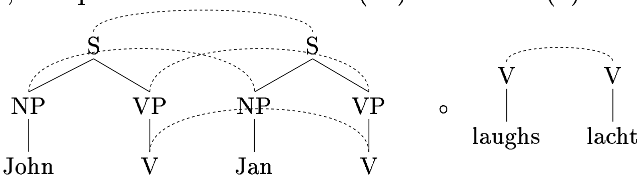
Given the trivial treebank in Figure 16.1, only the one translation pair can be processed. If we assume that the treebank fragments for the translation pair *Peter laughs* ⟷ *Piet lacht* are added to those in Figure 16.1 (these new fragments would be identical except for the different lexical material on the leaves of the tree fragments), then the two new translations in (3) can be handled on the basis of fragments already in the database:

(3) a. *John laughs* ⟷ *Jan lacht*

b. *Peter swims* ⟷ *Piet zwemt*

For example, one possible derivation of (3a) is that in (4):

(4)



That is, the V nodes in the rightmost tree pair can be substituted at the appropriate ⟨*source, target*⟩ V nodes in the leftmost tree pair. The full parse trees for the two sentences ensue, resulting in a *bona fide* translation given this treebank. The two translations in (3) will have a slightly smaller probability than the two original translation pairs given the presence of their full linked parse trees in the treebank, each of which has a probability of $\frac{1}{12}$. That is, there is one occurrence of the actual parse trees themselves, and there are 12 trees with root `cat=s`. However, given the trivial nature of the examples, there will be no other translation candidates output so each translation will have a probability of 1. This will, of course, not always be the case with more complex examples. For instance, in section 4 we examine in two experiments the situation where multiple translations are output in LFG-DOT with accompanying probabilities given the existence of both 'default' and 'specific' fragments in the database.

2.1.1 A Problem for DOT1. The DOT1 composition operation, based on leftmost substitution in the source trees, does not deal properly with translation cases where the word order differs significantly between two languages. Such an example is the *like* ⟷ *plaire* relation-changing case. As soon as the derivation in Figure 16.2 of the source sentence is arrived at, the desired linking of the English SUBJ with the French prepositional OBJ, and that of the English OBJ with the French SUBJ, are overridden by the composition operation of DOT1.

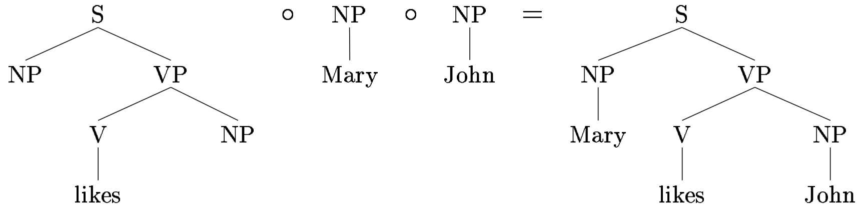


Figure 16.2. DOT derivation of the sentence *Mary likes John*

In this case the wrong translation of Figure 16.2 is derived, cf. Figure 16.3. It would appear that the adherence to leftmost substitution in the target given *a priori* leftmost substitution in the source is too strictly linked to the linear order of words, so that as soon as this deviates to any significant extent even between similar languages, DOT1 has a huge bias in favour of the incorrect translation. Even if the correct, non-compositional translation is achievable in such circumstances, 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.

2.2 DOT2: An Improved Model of Translation

Poutsma overcomes this problem in DOT2 by redefining the composition operation of DOT1 to operate on *pairs* of trees, rather than on single (source) trees. This new definition of composition ensures, among other things, that relation-changing cases such as *like* ⟷ *plaire* are handled correctly: instead of the wrong derivation in Figure 16.3 of the source sentence in Figure 16.2, we now obtain the correct translation *Jean plaît*

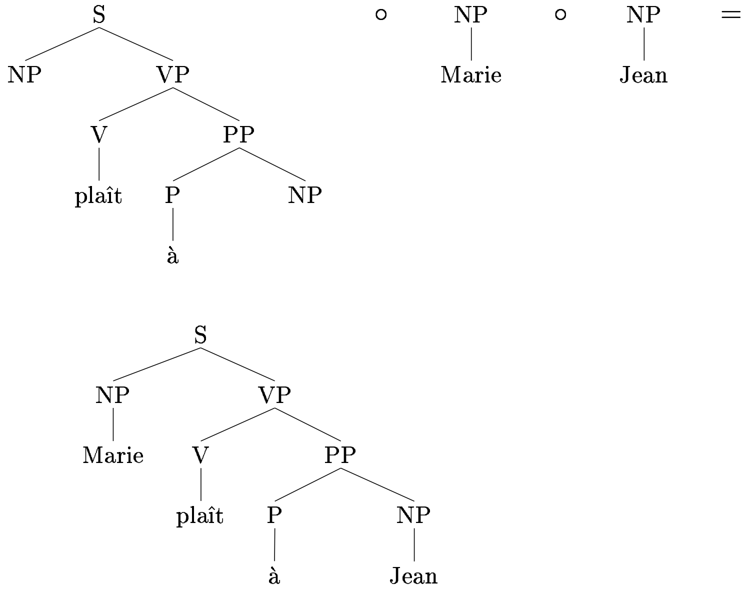


Figure 16.3. Wrong Translation of the sentence *Mary likes John* due to the DOT1 Composition Operation

à Marie. In DOT1, leftmost substitution in the source overrides the desired DOT-links between the French subject NP and the English object, as well as those between the French prepositional object and the English subject, given that composition is defined on the source tree *only.* Once pairs of trees are taken into account in DOT2, these links ensure the correct translation.

Way, 2001 shows that while DOT2 is able to handle certain 'hard' cases correctly, other examples, such as headswitching, are dealt with in a 'semi-compositional' manner. Consider the data in (5):

(5) a. DE: Johannes schwimmt gerne ⟷ EN: John likes to swim.
b. DE: Josef läuft zufällig ⟷ EN: Joseph happens to run.

Presupposing the derivation of a monolingual treebank constructed from the German examples in (5), with two different NPs, verbs and adverbs, eight sentences are possible and can receive analyses with associated probabilities with respect to that DOP corpus. However, only four of these possible sentences can receive translations in a DOT corpus, namely the examples in (5) as well as those in (6), by simple substitution of the alternate NPs into the respective subject slots:

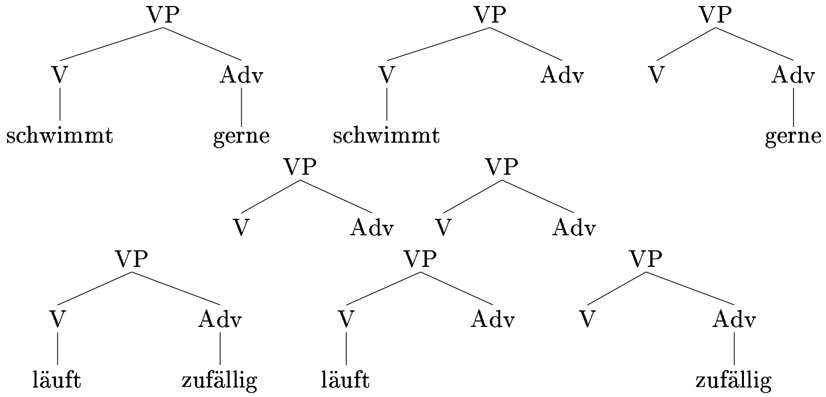


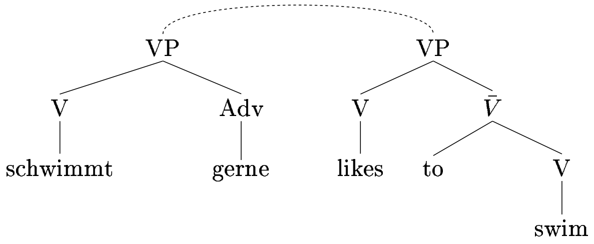
Figure 16.4. The monolingual DOP VP-fragments for a treebank built from the German examples in (5)

(6) a. DE: Josef schwimmt gerne ⟷ EN: Joseph likes to swim.

b. DE: Johannes läuft zufällig ⟷ EN: John happens to run.

The other four possible sentence pairs cannot be handled at all in a DOT treebank built from analyses of the strings in (5). This is due to the fact that the linked VP pairs are not broken down any further than at the root level. The contrast can be seen by examining the *schwimmt gerne* VP and its constituent DOP-fragments in Figure 16.4 and (7), which contains the single DOT linked VP pair:

(7)



We cannot draw a link between *schwimmt* and *swim* in (7) as they are not translationally equivalent: one is a finite verb and the other an infinitive. We cannot, therefore, express the basic translation relations as ⟨*gerne, likes to*⟩ and ⟨*zufällig, happens to*⟩. Given this restriction, the only way that the other sentence pairs can be handled is if there is some prior linked pair *läuft gerne* ⟷ *likes to run* as well as a prior instance of the linked pair *schwimmt zufällig* ⟷ *happens to swim.* This is because these linked VP pairs are handled non-compositionally in DOT2 between German and English, but the monolingual VPs are treated compositionally in DOP. Contrast this situation with a DOT treebank designed to translate these 8 German strings into Dutch. Our starting point could be the German strings in (5) with their Dutch translations, as in (8):

(8) a. DE: Johannes schwimmt gerne ⟷ NL: Johan zwemt graag.

b. DE: Josef läuft zufällig ⟷ NL: Josef loopt toevallig.

Given these simple transfer (i.e. 'word for word') examples, a DOT treebank would resemble much more closely the monolingual DOP treebanks from which it is derived for the respective sentences in (8), as *every* constituent part of the German strings corresponds exactly to a constituent part of the Dutch strings. In the DOT treebank these links are made explicit. When we have a headswitching case, however, it is apparent that both DOT models would translate the sentences correctly *iff* prior examples of linked headswitching VPs exist in the treebank. Such translations would receive extremely low probabilities with respect to the corpus in the normal case as they are built with a minimal degree of compositionality (substitution of subject NP, no other derivations being possible). As these examples only ever occur rarely, the chances of DOT2 managing to translate these in practice becomes significantly lower than might otherwise be expected, as we require not only the presence of the adverb, but also its occurrence to be correlated exactly with the verb in question for translation to succeed. We shall show that the LFG-DOT4 model of translation is able to provide generalized translation fragments which enable fully compositional translation in these cases, as required.

2.3 Boundary Friction in DOT

It is clear that DOT2 is an improvement on DOT1. DOT1 cannot always explicitly relate parts of the source language structure to the corresponding, correct parts in the target structure, so fails to translate correctly where source and target strings differ significantly with respect to word order. In DOT2, correct translations are obtained along

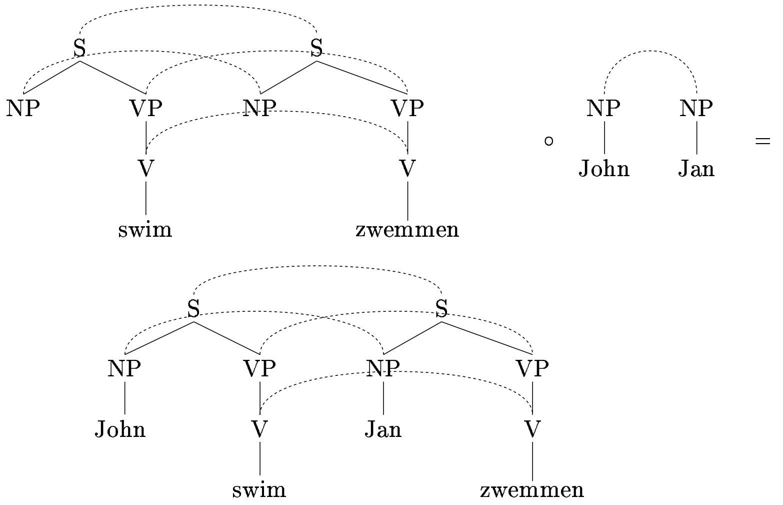


Figure 16.5. The Boundary Friction Problem in DOT

with some possible wrong alternatives. Notwithstanding the improved composition operation and probability model of DOT2, its ability to achieve the correct translation is compromised by a lesser amount of compositionality in the translation process. Given the small number of fragments playing a role in the derivation of some translations involving complex phenomena, almost the exact linked sentence pair may need to be present in order for a translation to be possible.

A further problem is that Poutsma's DOT models cannot distinguish ill-formed from well-formed input. For example, both DOT models of translation would permit the derivations in Figure 16.5. That is, with no stipulation on subject-verb agreement, it is perfectly legitimate in DOP-based models (and EBMT systems based purely on trees, e.g. Watanabe & Takeda, 1998, cf. Chapter 14, this volume), as well, of course, as systems where examples are stored as pairs of strings (e.g. Somers *et al.*, 1990) to combine a singular subject with a plural verb and end up with a well-formed object. In DOT, we end up with a translation which is well-formed given the corpus. This example illustrates that while the presence of category information alleviates the problem of boundary friction to a certain extent (compared to 'pure' example-based methods, for instance, where the examples are stored as strings with no linguistic information present at all), this continues to be problematic for DOT in certain circumstances. We show in Figure 16.7 that the analogous derivation in LFG-DOT would be deemed ungrammatical. That is, as soon as grammatical information is available via the accompanying f-structures, such a combination is impossible given the clash in NUM values for the subject and verb. Such ill-formed input can still be handled by relaxing grammaticality constraints such as these via *Discard*, but such translation pairs will be regarded as ungrammatical with respect to the corpus given their derivation via *Discard*; in DOT models, we have no such distinction.

2.4 Toward More Powerful Models

Finally, of course, translation systems which are based purely on phrase-structure (PS)-trees will ultimately not be able to handle certain linguistic phenomena. DOP-based approaches are necessarily limited to those contextual dependencies actually occurring in the corpus, which is a reflection of surface phenomena only. Purely context-free (CF) models are insufficiently powerful to deal with all aspects of human language.

Lexical Functional Grammar (Kaplan & Bresnan, 1982) is a theory of syntax which allows description of sentences at different levels—c-structure and f-structure.[2] The c-structure (*constituent* structure) is a PS-tree signifying the surface structure of the string, while the f-structure (*functional* structure) is an attribute-value matrix capturing the grammatical relations (in terms of arguments and adjuncts) inherent in the string. The mapping ϕ relates c-structure nodes onto elements of the f-structure. An example of the relevant structures for the sentence *John swims* is shown in Figure 16.6.

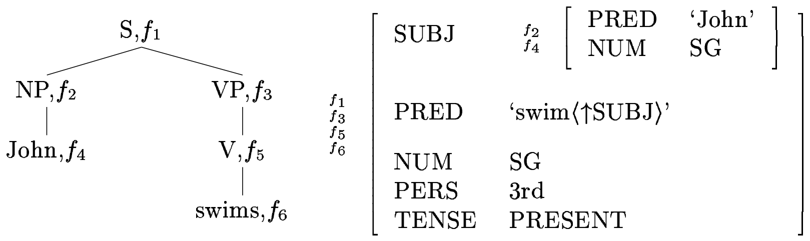


Figure 16.6. LFG Representations for the sentence *John swims*

The c-structure is built using the rules and lexical entries in (9):

[2] There is a third level, namely s-structure (semantic structure), and a second mapping σ, which relates f- and s-structures, but we omit s-structures from our translation models and will not discuss them further here.

(9)

$$\begin{array}{lcc} \text{S} \longrightarrow & \text{NP} & \text{VP} \\ & \uparrow\text{SUBJ}=\downarrow & \uparrow=\downarrow \end{array}$$

$$\begin{array}{lc} \text{VP} \longrightarrow & \text{V} \\ & \uparrow=\downarrow \end{array}$$

$$\begin{array}{ll} \textit{John}\text{: NP,} & \uparrow \text{PRED} = \text{`John'} \\ & \uparrow \text{NUM} = \text{SG} \end{array}$$

$$\begin{array}{ll} \textit{swims}\text{: V,} & \uparrow \text{PRED} = \text{`swim}\langle\uparrow\text{SUBJ}\rangle\text{'} \\ & \uparrow \text{NUM} = \text{SG} \\ & \uparrow \text{PERS} = \text{3rd} \\ & \uparrow \text{TENSE} = \text{PRESENT} \end{array}$$

We note that the functional attributes in the lexical entry for the NP *John* appear in the SUBJ f-structure slot of the sentence's (outermost) f-structure in Figure 16.6. This is due to the equation ↑SUBJ=↓ on the NP in the S-rule in (9), which essentially states that 'the SUBJ of what is above me (the S node) is filled in by the functional attributes of what is below me (*John*)'. Note also the ϕ-links (informally, parts of the f-structure linked from a c-structure node, cf. the f_n links in Figure 16.6, for instance) showing that the NP (f_2) and *John* (f_4) nodes are linked via ϕ to the SUBJ f-structure.

The annotation ↑=↓ on the VP node in the S-rule shows that this is the functional head of the sentence. The same relation can be seen with respect to the VP rule, where the verb bears the ↑=↓ annotation. Accordingly, we see the functional information of the verb appearing at the outermost level of the f-structure, with the semantic form ↑PRED='swim⟨↑SUBJ⟩' representing the subcategorization frame for *swims*. Again, note that the S, VP, V and *swims* nodes are ϕ-linked to the outermost, sentential f-structure.

We noted above that DOT cannot handle non-CF phenomena. 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, as seen in Figure 16.6, for example. With this in mind, 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.

3. LFG-DOT Models of Translation

Way, 2001 presents four possible LFG-DOT translation models, all of which use LFG-DOP as their language models. LFG-DOP models are defined using the same four parameters as in Tree-DOP. Its **representations** are 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. 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 fragments, thereby maintaining the fundamentals of c- and f-structure correspondence. 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.

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 (for instance, features such as TENSE are deleted when the main verb is removed, as this feature—in English, at least—is inextricably linked to the verbal PRED). *Frontier* operates as in Tree-DOP, deleting all subtrees of the selected frontier nodes. Furthermore, it deletes all ϕ-links of these removed nodes together with any semantic form corresponding to the same nodes. Finally, a third, new operation, *Discard*, provides generalized 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 ϕ-linked to some remaining c-structure node, 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, the **probability model** of LFG-DOP is again based on the relative frequency of a fragment, which in this case is a $\langle c, f \rangle$ pair. Bod & Kaplan give definitions of three possible competition sets from which fragments are sampled, depending on which point during the sampling stage the well-formedness conditions are invoked. These include the *Root* matching condition of DOP, and the following LFG conditions on grammaticality:

- *Uniqueness*: each attribute can have only one value;
- *Completeness*: all arguments of a local PRED must appear in the f-structure;

- *Coherence*: no arguments may appear in an f-structure which are not arguments of a local PRED.

Note that given the non-monotonic property of the Completeness check, this can only be enforced after all other validity sampling has taken place. The interested reader should consult Way, 2003 for details of the probability models for LFG-DOT.

3.1 LFG-DOT Model 1: Translation via τ

The first two LFG-DOT models propose the use of LFG's τ-equations to relate translation fragments between languages, the second in combination with γ, the function that links DOT source and target subtree fragments. Using τ-equations overcomes some of the problems of the DOT1 translation model. For instance, the LFG-MT solution (10) to the *like* ⟷ *plaire* relation-changing case can be availed of quite straightforwardly:

$$(10) \qquad \begin{array}{ll} \textit{like}: & (\tau \uparrow \text{PRED}) = \text{plaire} \\ & \tau(\uparrow \text{SUBJ}) = (\tau \uparrow \text{OBL}) \\ & \tau(\uparrow \text{OBJ}) = (\tau \uparrow \text{SUBJ}) \end{array}$$

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

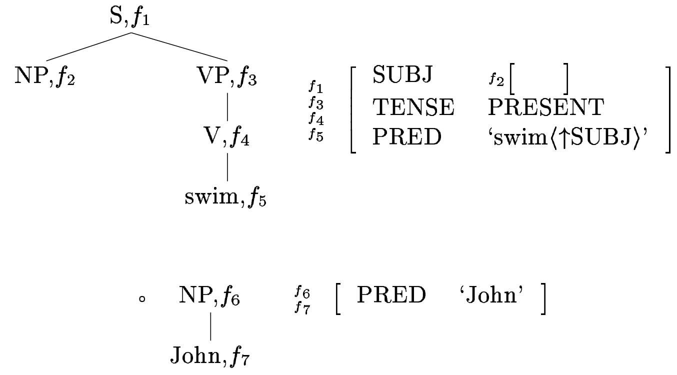


Figure 16.7. Robustness in LFG-DOP via *Discard*

In addition, DOP adds robustness to LFG-MT. LFG-DOT models of translation contain two monolingual LFG-DOP language models, so *Discard* can be run on both source and target sides. This means that LFG-DOT can cope with ill-formed or previously unseen input which LFG-MT would not be able to handle. Suppose that *John swim* is encountered as the source string, as in Figure 16.5. In LFG-DOP, this can only be interpreted if *Discard* relaxes certain constraints in the f-structures. One such derivation is shown in Figure 16.7. That is, the `SUBJ:NUM:PL` path will have been relaxed in the sentential f-structure, with the `NUM:SG` feature removed from the lower NP f-structure. The NP c-structure can be substituted at the NP node in the upper c-structure, and the f-structures unified. Given that such a derivation is possible only via *Discard*, it is regarded as ungrammatical with respect to the corpus, thereby maintaining a clear distinction between grammatical derivations permitted via *Root* and *Frontier*, and ungrammatical derivations.

Notwithstanding this, it is clear that LFG-DOT remains a robust system capable of handling unseen input, as required. The resultant f-structure would be input into the translation phase, which in LFG-DOT1 is quite simply the LFG-MT τ function. Taking the f-structure resulting from the derivation in Figure 16.7 as input, the appropriate τ-equations would build the corresponding target Dutch f-structure, which would be linked by the target language LFG-DOT model to the appropriate c-structure tree, as in Figure 16.8.

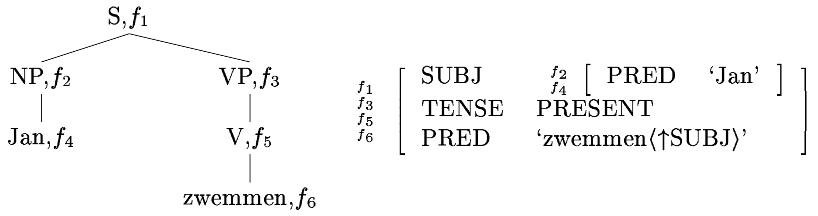


Figure 16.8. Robustness via *Discard* in LFG-DOT

The 'translation' of the ill-formed string *John swim* would therefore be *Jan zwemmen.* This shows that this particular DOT problem of boundary friction is resolved in LFG-DOT, owing to the presence of the syntactic f-structure information. We report further on the outstanding effect of boundary friction in LFG-DOT in section 4.

3.2 LFG-DOT Model 2: Translation via τ and γ

LFG-DOT2 requires the integration of the τ and γ mappings. Maintaining an f to f' translation engine in addition to γ adds more evidence in favour of any correct translations produced: even if these are not proposed as the most probable translations via τ, this function will only ever produce very few translation candidates (given the limited number of constraints that can play a part in the construction of translation candidates for any given source string), so we can guarantee in almost all cases that they are suggested as one of a small set of candidate translations. These can be compared to the best translation generated by γ and the highest ranking overall translation selected as output.

However, the τ mapping cannot always produce the desired translation (cf. Sadler *et al.*, 1989, Sadler *et al.*, 1990), so that most of the LFG-MT problems (notably, failure to deal successfully with certain headswitching examples, cf. Figure 16.9–Figure 16.11) are imported into both LFG-DOT1 and LFG-DOT2 models of translation.

In contrast, Way, 2001 shows that γ is always (depending, of course, on the coverage in the treebank) able to produce the correct translation, along with some possible wrong alternatives. The next two LFG-DOT models, therefore, abandon τ-equations and rely solely on γ to express the translation relation.

3.3 LFG-DOT Model 3: Translation via γ with Monolingual Filtering

The LFG-DOT3 model contains the DOT2 links between source and target c-structures, but with additional syntactic functional constraints which prevent ungrammatical structures such as Figure 16.5 from being formed (except via *Discard*, in much the same way as in Figure 16.7 above), thereby enabling truly grammatical translations to be output, as opposed to translations which are grammatical only 'with respect to the corpus'. The f-structure information can be seen, therefore, as useful for monolingual disambiguation in both source and target sides. Ill-formed or unknown input is still processable by running *Discard* over the set of linked source and target $\langle c$, LFG-DOP-$\phi, f\rangle$ fragments.

As an example, consider the *just* $\longleftrightarrow$ *venir de* headswitching case. In terms of LFG-DOT3, the translation relation is shown in Figure 16.9.[3] The semantically equivalent source and target c-structures are linked via γ only at the VP level (cf. the dotted lines linking the $\langle$*source, target*$\rangle$ VP nodes). We do not consider *fell* to be semantically equivalent to *tomber* owing to their different FIN(ite) values, added to the fact that *fell* has a TENSE value whilst *tomber* does not. Hence this translation fragment can only be reused by substituting this pair together with associated singular NP subjects at the appropriate nodes in an S-linked fragment.

[3] The solid line linking the French SUBJ with the XCOMP SUBJ indicates that the SUBJ at the outermost f-structure re-enters into the XCOMP SUBJ slot, i.e. it is the 'understood' subject of the infinitival construction.

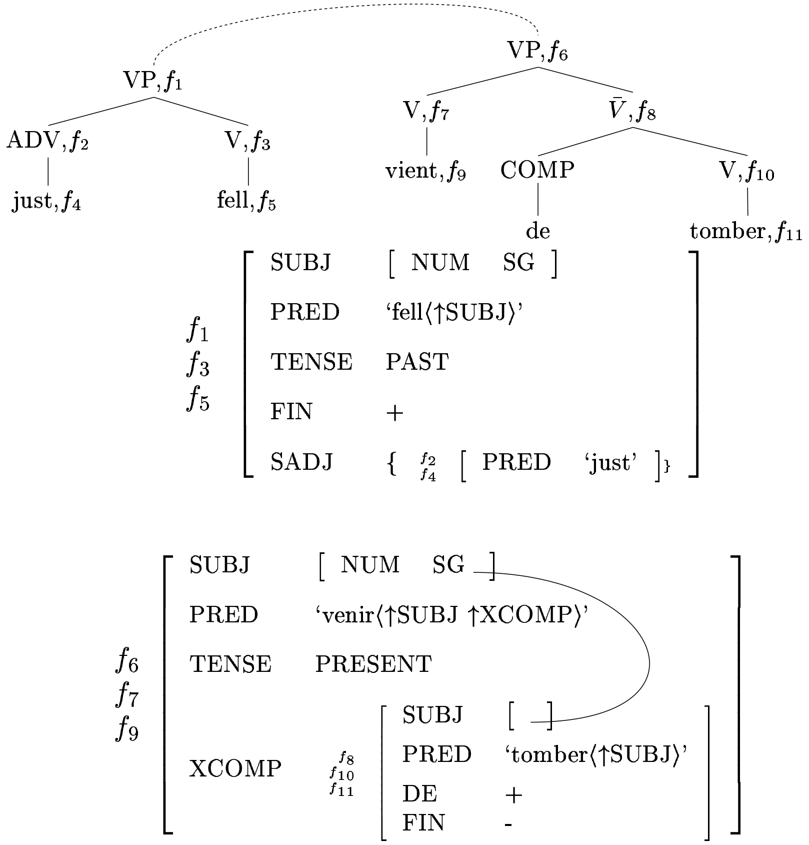


Figure 16.9. The *just* ⟷ *venir de* headswitching case in LFG-DOT3

As an example, we could combine the fragments for *the man* and *just fell*, and produce the translation *l'homme vient de tomber*, but we cannot generalize the fragments in Figure 16.9 to enable other verbs to appear in the context of *just* and *venir de.* The only way in which *just left* ⟶ *vient de partir* can be handled, for example, is if such a linked fragment already existed in the treebank—the fragments in Figure 16.9 would be of no use whatsoever in dealing with this new translation case. In this respect, as with DOT2, this LFG-DOT3 model continues to suffer from limited compositionality. We address this concern further in the next section which describes the LFG-DOT4 model.

3.4 LFG-DOT Model 4: Translation via γ and ‘Extended Transfer’

In the previous section, we observed that the outstanding problem with LFG-DOT3 is its retention of the DOT2 problem of limited compositionality. Returning to the *just* $\longleftrightarrow$ *venir de* headswitching case in Figure 16.9, we would like to be able to ‘relax’ some of the constraints in order to map ⟨*fell,tomber*⟩ to make these linked fragments more general, and hence more useful. In so doing, we would remove this problem of limited compositionality.

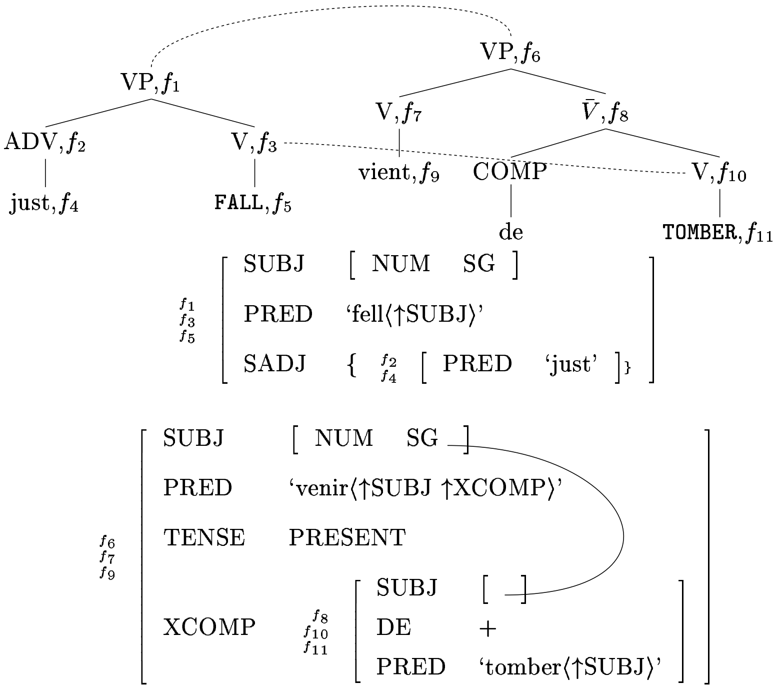


Figure 16.10. Lemmatization in LFG-DOT4

In LFG-DOT4, the basic translation relation is expressed by γ, as in LFG-DOT3. In LFG-DOT4, however, there is a second application of *Discard*, by which ‘lemmatized’ forms are arrived at on which ‘extended transfer’ can be performed. *Discard* relaxes constraints in order to produce a set of generalized fragments with the potential to deal with ill-formed or unknown input. Once the TENSE and FIN features have been relaxed on the lowest verbs in both fragments in Figure 16.9, they

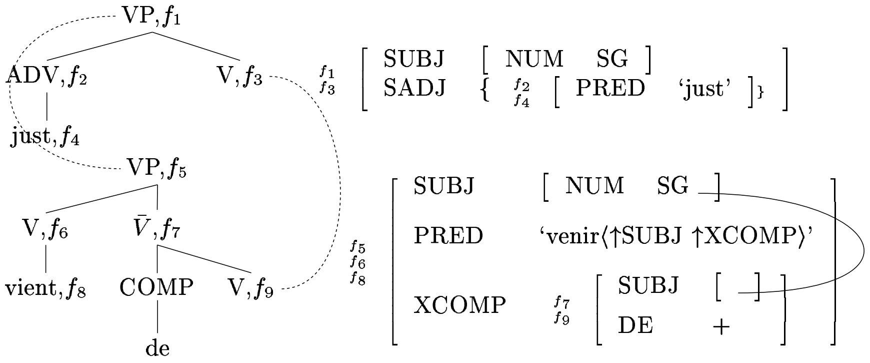


Figure 16.11. Generalized form of the *just* ⟷ *venir de* translation relation in LFG-DOT4

can be regarded as translationally equivalent. Given this, ⟨*fell,tomber*⟩ are linked and lemmatized, as in Figure 16.10.[4]

Now that ⟨FALL,TOMBER⟩ are linked, they can be deleted to produce the generalized form of the translation relation in Figure 16.11, as required. If fragment pairs such as those in Figure 16.11 prove subsequently to be of use in combining with other fragments, any resultant translation will be marked as ungrammatical with respect to the corpus, given that *Discard* was used in its derivation. Nevertheless, even if we restrict the impact of *Discard* on the probability space (cf. Way, 2001, Way, 2003), in order to ensure a preference for well-formed analyses derived via *Root* and *Frontier* over those produced by *Discard*, such translations will receive *some* probability, whereas the semi-compositional variants from which they were derived may not be able to produce *any* translation in practice. That is, the linked pair of fragments in Figure 16.11 can now be used to translate *The man just left* ⟶ *L'homme vient de partir*, which was not the case with the LFG-DOT3 fragments in Figure 16.9.

4. Boundary Friction in LFG-DOT: Two Experiments

Both DOT and LFG-DOT have strict definitions of fragment boundaries. We showed that whereas DOT cannot distinguish well-formed from ill-formed structures, LFG-DOT has an intuitively correct notion

[4] We ignore here the issue of the lemmatization of *vient*, cf. Way, 2001:224f. for further discussion.

of grammaticality. Nevertheless, the thorny issue of boundary friction does raise its head in LFG-DOT to a degree.

All MT systems have to decide what are legitimate translation candidates. In most rule-based systems, default rules are differentiated from specific rules, with the former applying only in those cases where a specific rule cannot. Watanabe, 1994 (cf. Chapter 14, this volume) discusses the problem of boundary friction (he calls it 'example interference'), and provides a method of distinguishing exceptional from general examples in EBMT on the basis of *similarity* of examples (cf. Sato & Nagao, 1990, who use a similar technique based on thesaurus relations). Once patterns are identified as general, exceptional or neutral, some of the side-effects of boundary friction may be overcome.

Not all rule-based systems can prevent the output of a wrong, compositional translation once a specific translation has been obtained. For instance, in LFG-MT, satisfying the requirement that only possible translations are produced is problematic where the translation of a lexical head is conditioned in some way by one of its dependants, as in (11):

(11) commit suicide $\longleftrightarrow$ se suicider

The problem is that in these cases, suppressing the wrong, compositional translation in LFG-MT is impossible. For instance, we require the default rules in (12):

(12) a. commit $\longleftrightarrow$ commettre

b. suicide $\longleftrightarrow$ suicide

Such rules are expressed in LFG-MT by the lexical entries in (13):

(13)

commit: $(\tau \uparrow \text{PRED}) = \text{commettre}$
$\tau(\uparrow \text{SUBJ}) = (\tau \uparrow \text{SUBJ})$
$\tau(\uparrow \text{OBJ}) = (\tau \uparrow \text{OBJ})$

suicide: $(\tau \uparrow \text{PRED}) = \text{suicide}$

That is, the default translation (via τ) of *commit* is *commettre*, in which case its SUBJ and OBJ NPs are translated straightforwardly as the SUBJ and OBJ respectively of *commettre*. Similarly, the default translation of the English word *suicide* in French is *suicide*. These entries show how *commit* and *suicide* are to be translated under normal ('word for word') circumstances, such as in (14):

(14) a. John commits a crime ⟷ Jean commet un crime.

b. Suicide is tragic ⟷ Le suicide est tragique.

Nevertheless, given the default, compositional entries in (13), LFG-MT produces the wrong translation in (15):[5]

(15) John commits suicide ⟷ *Jean commet le suicide.

LFG-MT can, however, derive the correct translation *John se suicide* in such cases via the solution in (16):

(16) *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.* The selective use of constraining equations enables correct translations to be derived which would only be possible in other systems by tuning. Nevertheless, the point remains that in LFG-MT we would get both translations here, i.e. a correct one and a wrong one, since it is not possible to enforce the requirement that specific rules ought to override the default translational rules where applicable. Furthermore, these translations would be output with no ranking, so linguistic expertise in both source and target languages may be required in order to select the better translation.

4.1 Experiment 1

We tested the issue of default versus specific translations in LFG-DOT3. We produced an LFG-DOT3 treebank containing all the linked fragments from the sentences in (17):

[5] Note that the rules in (12) are *bona fide* translation rules that any rule-based English-French MT system will require. It is, therefore, the task of the French generation component to explicitly rule out the incorrect translation in (15), *not* the transfer component.

(17) a. Le suicide est tragique ⟷ Suicide is tragic.

b. Jean commet le crime ⟷ John commits the crime.

c. Jean commet le meurtre ⟷ John commits the murder.

d. Jean dort ⟷ John sleeps.

e. Marie se suicide ⟷ Mary commits suicide.

f. Marie commet un attentat ⟷ Mary commits an attack.

g. Marie commet la faute ⟷ Mary commits the mistake.

h. Pierre commet un arbitre ⟷ Peter nominates an arbitrator.

i. Pierre commet une erreur ⟷ Peter commits an error.

j. Pierre commet une injustice ⟷ Peter commits an injustice.

Here there are seven instances of *commettre* (six of which translate as *commit*) as opposed to only one instance of *se suicider*.

Before examining results obtained with LFG-DOT, it is insightful to point out that in the monolingual French LFG-DOP (no *Discard*) treebank built from the French strings in (17), *Marie se suicide* is preferred about 2.6 times over the compositional alternative *Marie commet le suicide*. If these were the output translations, then ranking them against one another would favour *Marie se suicide* with a probability of 0.72 while *Mary commet le suicide* has a probability of 0.28. In the LFG-DOT3 treebank produced from the English and French sentences in (17), the specific translation is preferred even more than in the French monolingual LFG-DOP treebank. We set out to test the weight of the specific over the compositional translation for the sentences in (18):

(18) a. John commits suicide ⟷ Jean se suicide

b. Mary commits suicide ⟷ Marie se suicide

c. John commits suicide ⟷ *Jean commet le suicide

d. Mary commits suicide ⟷ *Marie commet le suicide

Translation (18a) can be built using the three derivations in Figure 16.12.[6] (18b) has the additional derivation of the full trees (and accompanying f-structures) for this sentence pair. The probabilities of (18a-b) are shown in (19):

[6] We have omitted the accompanying f-structure fragments for reasons of space.

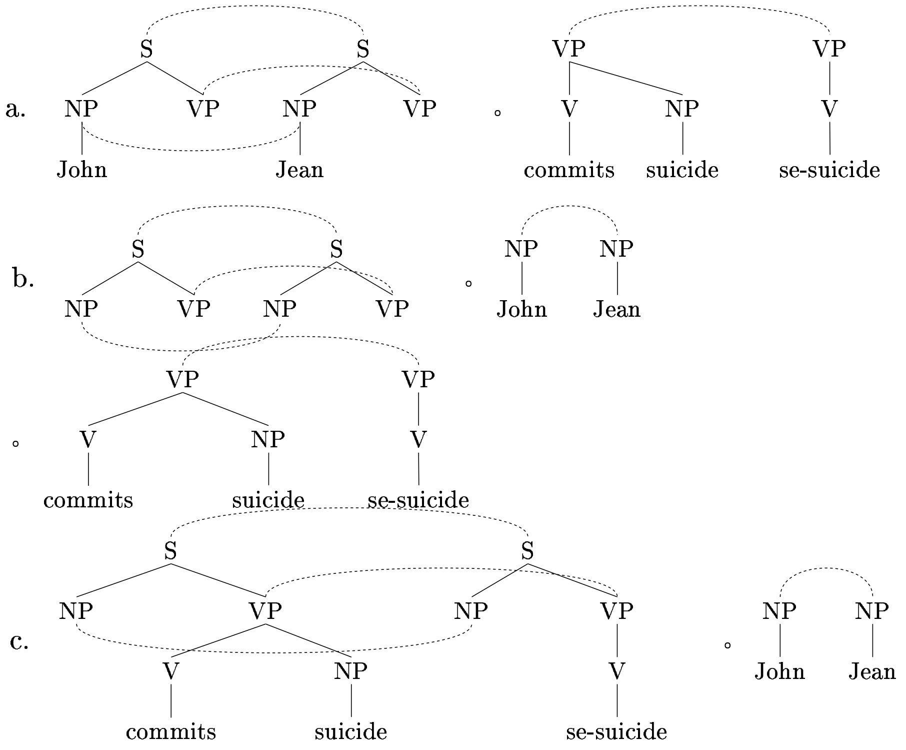


Figure 16.12. C-structure derivations for *John commits suicide* ⟺ *Jean se suicide*

(19) a. P(John commits suicide $\longleftrightarrow$ Jean se suicide) $= 0.000705$ ($\simeq \frac{1}{1419}$)

b. P(Mary commits suicide $\longleftrightarrow$ Marie se suicide $= 0.006229$ ($\simeq \frac{1}{161}$)

For each of the translations in (18c-d) there are 7 derivations with total probability 0.000501 ($\simeq \frac{1}{1998}$). Now we can rank each translation with respect to the other in (20):

(20) a. P(John commits suicide $\overset{T}{=}$ Jean se suicide) $= 705/1206 = 0.5846$

b. P(John commits suicide $\overset{T}{=}$ Jean commet le suicide) $= 501/1206 = 0.4154$

c. P(Mary commits suicide $\overset{T}{=}$ Marie se suicide $= 6229/6750 = 0.923$

d. P(Mary commits suicide $\overset{T}{=}$ Marie commet le suicide) $= 521/6750 = 0.077$

Here $\mathcal{W}\overset{T}{=}\mathcal{T}$ means that $\mathcal{T}$ is a translation of a word string $\mathcal{W}$. Therefore we can see that for *John commits suicide*, the correct, specific translation is about 1.4 times more likely than the wrong, default, compositional translation, whereas for *Mary commits suicide* the specific translation is preferred about 12 times more than the default translation. We see in (20) the dominance of the exact linked translation pair over the alternative translation. The presence of the exact translation (17e) is insufficient to explain the preference for the specific translation of *John commits suicide*: despite the presence of six *commit* $\longleftrightarrow$ *commettre* examples in (17) compared to only the single instance of *commits suicide* $\longleftrightarrow$ *se suicide*, the specific translation is nonetheless preferred.

4.2 Experiment 2

How many more times empirically we can expect to see *commit* $\longleftrightarrow$ *commettre* compared to *commits suicide* $\longleftrightarrow$ *se suicide*? In the LOB Corpus, 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* covering 1986-1993, there are just 106 instances of *se suicider* (including its morphological variants). There will, of course, be many more instances of *commettre*. Given occurrences of *suicide* as an NP in French corpora, it is not an unreasonable hypothesis to expect that wrong translations such as (15) will be much more probable than those derived via the specific rule. However, this hypothesis is shown to be inaccurate in the above experiment.

Furthermore, it is clear from the results in (20) that a ratio of 6:1 is insufficient to achieve a bias in favour of the wrong, compositional translation in LFG-DOT. Running a new experiment with a treebank built from 5 instances of each translation pair (17a-d) and (17f-j) and just the one instance of (17e), making a total of 46 translation pairs in all, produces the results in (21):

(21) a. P(John commits suicide $\overset{T}{=}$ Jean se suicide) = 132/635 = 0.208

b. P(John commits suicide $\overset{T}{=}$ Jean commet le suicide) = 503/635 = 0.792

c. P(Mary commits suicide $\overset{T}{=}$ Marie se suicide = 1206/1758 = 0.686

d. P(Mary commits suicide $\overset{T}{=}$ Marie commet le suicide) = 552/1758 = 0.314

Now, with 30 instances of *commit* ⟷ *commettre* and only the one *commits suicide* ⟷ *se suicide* example, we see that the wrong, default, compositional translation for *John commits suicide* is now preferred by about 3.8 times, but the presence of the exact translation (17e) maintains the preference for the specific translation for *Mary commits suicide* by about 2.2 times. Consequently we can see that it will take many more instances of *commit* ⟷ *commettre* before the specific translation for *Mary commits suicide* is outranked by the wrong, compositional alternative.

5. Conclusions and Future Work

Models of translation based on DOP and LFG-DOP translate new strings on the basis of linked ⟨*source, target*⟩ fragments already located in their databases. Accordingly, such systems may be viewed as example-based systems.

We described the DOT models of translation based on DOP. DOT1 is not guaranteed to produce the correct translation when this is non-compositional and considerably less probable than the default, compositional alternative. DOT2 addresses the failings of DOT1 by redefining the composition operation. In contrast to DOT1, DOT2 cannot fail to produce correct candidate translations, along with some possible wrong alternatives, depending of course on the corpus from which fragments are derived. Despite the presence of syntactic information in the tree-structure fragments, we showed that both DOT models continue to suffer from the problem of boundary friction in cases where singular and plural fragments are combined.

We also described a number of new hybrid models of translation which use LFG-DOP as their language models. The first, LFG-DOT1, imports the τ-equations from LFG-MT as the translation relation. LFG-DOT1 improves the robustness of LFG-MT through the use of the LFG-DOP *Discard* operator, which produces generalized fragments by discarding certain f-structure features. It can, therefore, deal with ill-formed or previously unseen input where LFG-MT cannot. Unsurprisingly, however, all of the other problems of LFG-MT are maintained in LFG-DOT1.

Given this, we augmented LFG-DOT1 with the γ function from DOT2 to give an improved model of translation. LFG-DOT2 maintains the τ translation relation to increase the chances of the correct translation being produced. Nevertheless, given that the τ-equations fail to derive the correct translation in all cases, we omitted the τ translation relation from our subsequent models.

LFG-DOT3 relies wholly on γ to express the translation relation, and uses f-structure information purely for monolingual filtering. The presence of this functional information prevents the formation of certain ill-formed structures which can be produced in DOT. LFG-DOT models, therefore, have a notion of grammaticality which is missing from DOT models. While both DOT and LFG-DOT contain strict notions of boundary definition, DOT allows the output of structures which are well-formed according to the corpus, but which are syntactically ungrammatical. The definition of well-formedness in LFG-DOT, in contrast, corresponds exactly to our understanding of grammaticality in the usual linguistic sense. However, both DOT2 and LFG-DOT3 models suffer from limited compositionality, so that in some cases the minimal statement of the translation relation is impossible.

LFG-DOT4 adds an 'Extended Transfer' phase to LFG-DOT3 by producing lemmatized forms using a second application of *Discard.* This extension overcomes the problem of limited compositionality, enabling the statement of the translation relation in an intuitive, concise fashion.

Finally, we demonstrated that LFG-DOT models of translation suffer less from the problem of boundary friction than DOT models given the presence of the additional syntactic f-structure information. In addition, we showed in two small experiments that despite attempting to 'load the dice' in favour of the wrong, compositional translation over the correct, specific alternative, LFG-DOT continues to translate in a robust fashion while maintaining a preference for the correct translation.

The work described here and in Way, 2001 uses as its evaluation metric the ability to cope with 'hard' translation cases, such as relation-changing, cf. Figure 16.2 and Figure 16.3, and headswitching, cf. (5)–(8). Special LFG-DOT corpora such as those derived to test the effect of boundary friction in section 4 needed to be created. The translation effects examined here need to be tested further on larger corpora, and the work of Frank *et al.*, 2001 and Cahill *et al.*, 2002 on semi-automatic derivation of LFG corpora from treebank resources would appear promising in this regard.

Furthermore, the hypotheses developed in Way, 2001 need to be further explored with respect to simpler translation data, such as *John swims* sentences. Different probability models will also be evaluated (cf. Bonnema *et al.*, 2000), as will the possibility of pruning the search space, by cutting down the number of fragments produced (cf. Sima'an, 1999) in order to improve the efficiency of the models proposed.

Acknowledgments

The author wishes to thank the anonymous referees whose comments led to improvements in this paper. All remaining errors are the responsibility of the author.

References

Abeillé, A., Y. Schabes and A.K. Joshi. 1990. Using Lexicalised Tags for Machine Translation. In *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, **3**:1–6.

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

Bod, R. and R. Kaplan. 1998. A Probabilistic Corpus-Driven Model for Lexical-Functional Analysis. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, Canada, **1**:145–151.

Bonnema, R., P. Buying and R. Scha. 2000. Parse Tree Probability in Data-Oriented Parsing. In *CICLING: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, pp.219–232.

Cahill, A., M. McCarthy, J. van Genabith and A. Way. 2002. Automatic Annotation of the Penn-Treebank with LFG F-Structure Information. In *Proceedings of the LREC Workshop on Linguistic Knowledge Acquisition and Representation: Bootstrapping Annotated Language Data*, Las Palmas de Gran Canaria, Spain, pp.8–15.

Frank, A., J. van Genabith and A. Way. 2001. Treebank vs. X-BAR based Automatic F-Structure Annotation. In M. Butt and T. Holloway-King (eds.) *LFG-2001: Proceedings of 6th International Conference on Lexical Functional Grammar*, CSLI Publications, Stanford, CA., pp.127–146.

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, MA., pp.173–281.

Kaplan, R., K.. Netter, J. Wedekind and A. Zaenen. 1989. Translation by Structural Correspondences. In *Proceedings of the Fourth Conference of the European Association for Computational Linguistics*, Manchester, England, pp.272–281.

Mima, H, H. Iida and O. Furuse. 1998. Simultaneous Interpretation Utilizing Example-Based Incremental Transfer. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, Canada, pp.855–861.

Poutsma, A. 1998. Data-Oriented Translation. In *Computational Linguistics in the Netherlands: Ninth CLIN Meeting*, Leuven, Belgium, [no page numbers].

Poutsma, A. 2000. *Data-Oriented Translation: Using the Data-Oriented Parsing framework for Machine Translation*, MSc Thesis, University of Amsterdam, Amsterdam, The Netherlands.

Sadler, L., I. Crookston, D. Arnold and A. Way. 1990. LFG and Translation. In *Third Conference on Theoretical and Methodological Issues in Machine Translation*, Austin, TX., pp.121–130.

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, UK.

Sato, S. and M. Nagao. 1990. Towards Memory-based Translation. In *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, **3**:247–252.

Schäler, R. 2001. Beyong Translation Memories. In M. Carl and A. Way (eds.) *Proceedings of the Workshop on Example-Based Machine Translation, MT Summit VIII*, Santiago de Compostela, Spain, pp.49–55.

Sima'an, K. 1999. *Learning Efficient Disambiguation*, PhD Thesis, University of Utrecht, Utrecht, The Netherlands.

Somers, H. 1999. Example-based Machine Translation. *Machine Translation* **14**(2):113–157 (cf. Chapter 1, this volume).

Somers, H., I. McLean and D. Jones 1994. Experiments in Multilingual Example-Based Generation. In *CSNLP-94: 3rd Conference on the Cognitive Science of Natural Language Processing*, Dublin, Ireland, [no page numbers].

Somers, H., J. Tsujii and D. Jones. 1990. Machine Translation without a source text. In *COLING-90, Papers Presented to the 13th International Conference on Computational Linguistics*, Helsinki, Finland, **3**:271–276.

Watanabe, H. 1992. A Similarity-Driven Transfer System. In *Proceedings of the fifteenth [sic] International Conference on Computational Linguistics, COLING-92*, Nantes, France, **2**:770–776

Watanabe, H. 1994. A Method for Distinguishing Exceptional and General Examples in Example-based Transfer Systems. In *COLING-94: The 15th International Conference on Computational Linguistics*, Kyoto, Japan, pp.39–44.

Watanabe, H. and K. Takeda. 1998. A Pattern-based Machine Translation System Extended by Example-based Processing. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal, Quebec, Canada, **2**:1369–1373.

Way, A. 2001. *LFG-DOT: A Hybrid Architecture for Robust MT*, PhD Thesis, University of Essex, Colchester, UK.

Way, A. 2003. Machine Translation using LFG-DOP. In R. Bod, R. Scha and K. Sima'an (eds.) *Data-Oriented Parsing*, CSLI, Stanford, CA.