Machine Translation using LFG-DOP

Andy Way

CENTER FOR THE STUDY
OF LANGUAGE
AND INFORMATION

Abstract

This chapter describes how LFG-DOP (Bod & Kaplan, 1998) can serve as a novel hybrid model for Machine Translation (MT), LFG-DOT (Way, 2001), which promises to improve upon DOT and LFG-MT systems. We propose four models of translation which use LFG-DOP as their language models. The first model, LFG-DOT1, uses the conventional transfer rules of the correspondence-based model of translation (LFG-MT: Kaplan *et al.*, 1989) as the mapping between source and target f-structures. LFG-DOT2 integrates the τ-equations of LFG-MT with the DOT2 c-structure links (Poutsma 1998; 2000). The final two models eschew τ-equations in favour of the DOT2 links, but these models use the additional f-structure information for monolingual filtering to significantly improve the DOT notion of grammaticality. LFG-DOT4 produces generalized translation fragments from those derived in LFG-DOT3, which like DOT2, suffers from limited compositionality when confronted with certain data. This amendment to LFG-DOT4 enables the statement of translation relations in an intuitive, concise fashion.

1 Data-Oriented Translation

Poutsma (1998; 2000; cf. chapter 21, this volume) has developed a model of translation based on Tree-DOP—Data-Oriented Translation (DOT). There are two different versions of DOT. DOT1 does not deal properly with translation cases where the word order differs significantly between two languages (e.g. the *like* $\longleftrightarrow$ *plaire* relation-changing case), given that the DOT1 composition operation is based on leftmost substitution in the source trees only. In such cases, wrong translations are obtained, as in (1):[1]

(1) Mary likes John $\longleftrightarrow$ *Marie plaît à Jean (cf. Jean plaît à Marie)

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.

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

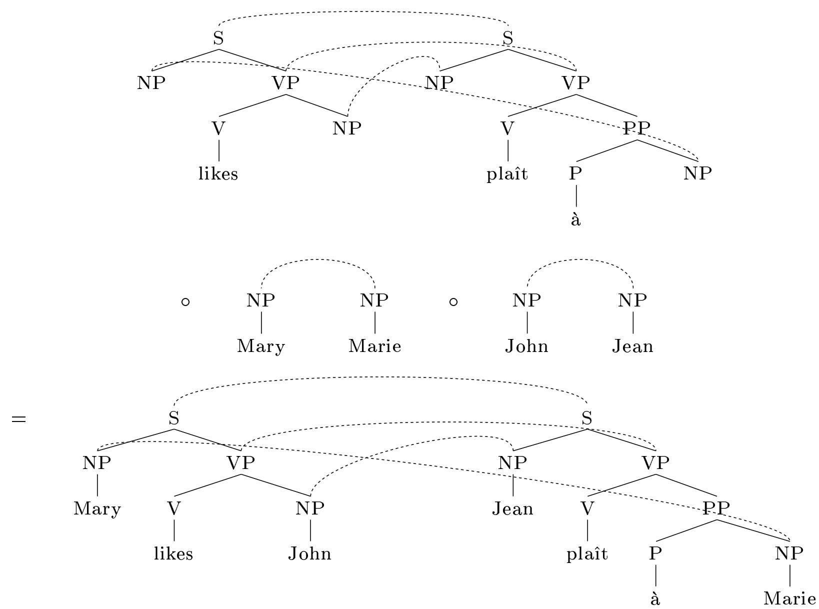
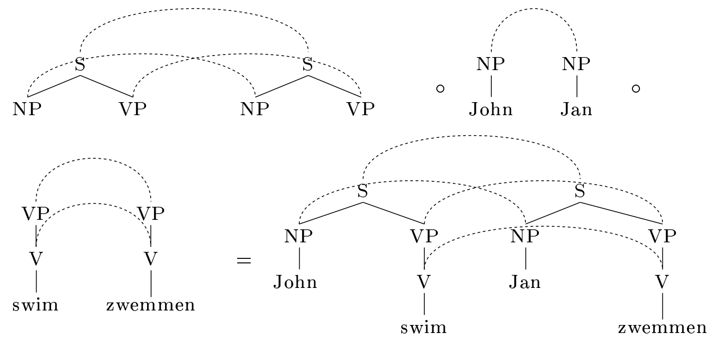


FIGURE 1 Correct derivation of *Mary likes John* ⟷ *Jean plaît à Marie* in DOT2

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, as in Figure 1. Despite leftmost substitution of *Mary* at the subject NP node of the source tree, we are no longer compelled to substitute the linked node *Marie* at the subject NP node of the target tree, as we were in DOT1. Note that a link cannot be drawn between *likes* and *plaît* as they are not translationally equivalent.

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 with some possible wrong alternatives. Way (2001) points out 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 (2):

(2)



That is, with no stipulation on subject-verb agreement, it is perfectly legitimate in DOP-based models 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. We show in (18) that the analogous derivation in LFG-DOT is deemed ungrammatical. That is, as soon as grammatical information is available via the accompanying f-structures, such a combination would be 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* (cf. chapter 16), 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.

In addition, 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. We will base our discussion on the data in (3):

(3) 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 (3), 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 (3) as well as those in (4), by simple substitution of the alternate NPs into the respective subject NP slots:

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

The other four sentence pairs in (5) cannot be handled at all in a DOT treebank built from the strings in (3):

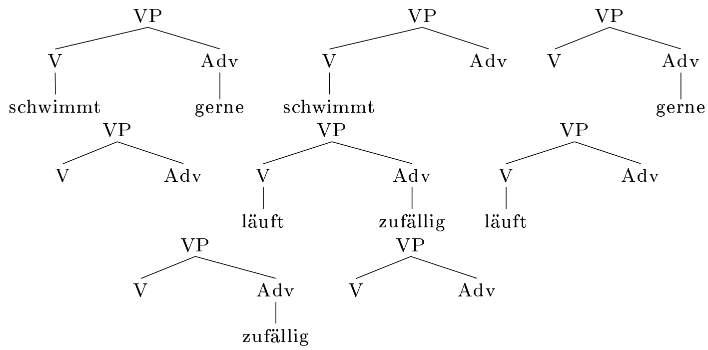
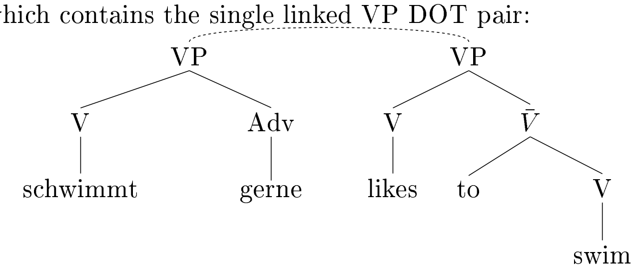


FIGURE 2 The monolingual DOP VP-fragments for a treebank built from the German examples in (3)

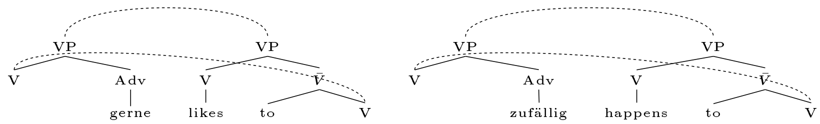
(5) a. DE: Johannes läuft gerne $\longleftrightarrow$ EN: John likes to run.
b. DE: Josef schwimmt zufällig $\longleftrightarrow$ EN: Joseph happens to swim.
c. DE: Josef läuft gerne $\longleftrightarrow$ EN: Joseph likes to run.
d. DE: Johannes schwimmt zufällig $\longleftrightarrow$ EN: John happens to swim.

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 2 and (6), which contains the single linked VP DOT pair:

(6)



In his DOT systems, Poutsma states that only semantically equivalent trees may be linked. Two trees T_1 and T_2 are semantically equivalent *iff* T_1 can be replaced by T_2 without loss of meaning. If it is true monolingually that a finite verb cannot be replaced by an infinitive without a loss of meaning resulting, the same must be true across languages. We cannot, therefore, link *schwimmt* and *swim* in (6). Accordingly, we cannot produce the fragments in (7), as we might otherwise wish to do, in order to describe the basic translation relations in (3).

(7) 

In such circumstances, the only way that the sentence pairs in (5) can be handled is if the linked pairs *läuft gerne* ⟷ *likes to run* and *schwimmt zufällig* ⟷ *happens to swim* already exist in the database. 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 (3) 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. For instance, let us add the sentence pairs with all other resultant linked fragments in (9) to those in (3):

(9) a. DE: Peter läuft gerne ⟷ EN: Peter likes to run.
b. DE: Markus schwimmt zufällig ⟷ EN: Mark happens to swim.

This now allows all 8 German strings in (3)–(5) to be correctly translated into English by substitution of the relevant subject NPs in this trivial corpus. They 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). 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. The chances of such a co-occurrence are small, but we suppose that using larger corpora will enable the translation to be correctly rendered. 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.

1.1 'Extended' DOT and Limited Compositionality

Despite the advocation of the principle of semantic equivalence, Poutsma *does* permit linkages such as ⟨*schwimmt, swim*⟩ in (3). We maintain that this is wrong, as such linkages violate the principle of semantic equivalence. Of course, one might instead choose to relax this constraint altogether, but in this case one would be abandoning the idea of compositional translation in its entirety: any source node could in principle be linked to any target node, which would obviously lead to a poor model of translation, with correspondingly poor results.

One might, however, consider that the use of features would provide a solution to the problem of limited compositionality. For instance, with respect to (6), one might relabel the V nodes with more fine-grained categories such as V_3rd_person in the German tree, and V_Infinitive in the English tree, link these nodes, and 'solve' the problem in DOT. There are a number of problems with this approach: (i) firstly, despite the trend nowadays for specific categories such as these in resources like the *Penn Treebank*, generalisations are lost to a degree; (ii) in our LFG-DOT models, we advocate the relaxation of constraints via *Discard* in order to produce generalised fragments which improve the robustness of LFG-DOT. In our system, this is done in a clean, principled way—*schwimmt* and *swim* could only be linked via 'extended transfer' in LFG-DOT4 (and not in any other LFG-DOT model), as the bundles of f-structure constraints associated with those nodes clearly differ. If such syntactic information were instead buried in complex category names, it is far from clear how such constraints may be relaxed. Furthermore, it is apposite here to reprise Johnson's (1988) work on Attribute-Value Grammars, which considered the difference between a representation and the constraints *underlying* that representation. Prior to this work, it is fair to say that many linguists had clouded this issue. Nevertheless, there is a clear distinction between linguistic objects and the language used to describe the constraints. For instance, Johnson observes that:

> "In this system attribute-value structures play only one role: they are defined in order to give a semantics for the language that describes them. None of the algorithms in the following chapters actually constructs an attribute-value structure: instead, they operate on descriptions of attribute-value structures to determine the existence and the probabilities of those structures"(*op cit.*, p.12).

For Johnson, therefore, the underlying linguistic constraints are what is important; the structures that can be built from them are merely an encoding of these constraints which make it easier to visualize what they refer to. Similarly, while we are arguing specifically here in favour of an approach to translation founded on LFG-DOP, in the wider context we are advocating a tree-based approach with monolingual syntactic constraints. Whether that be an LFG-based approach, or some such other proposal, is not ultimately of importance. So rather than having more complex category names, one could envisage an alternative 'extended DOT model' where the 'normal' monadic category names are maintained, but with additional features which are accessible for deletion by a generalisation process, if required. If this were equivalent to an 'augmented CFG', there would still be certain elements of language that would not be processable by such a model.[2]

Translation systems which are based purely on 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. 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. Lexical Functional Grammar (LFG: Kaplan & Bresnan, 1982), however, is known to be beyond context-free. It can capture and provide representations of linguistic phenomena other than those occurring at surface structure. With this facility 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 not available to models based solely on LFG.

[2]It is an interesting question as to whether there are any translation cases which require context-sensitive power. Perhaps the nearest approximations to the DOP approach are Probabilistic Tree Adjoining Grammar (Resnik 1992) and Stochastic Lexicalized Tree Adjoining Grammar (Schabes 1992). Such TAG-based approaches require adjunction (i.e. CS-power), rather than merely tree substitution, in cases such as the French negation elements *ne ... pas*. Nevertheless, DOP-based approaches will produce tree fragments which contain both *ne* and *pas*, so this collocation would seem to be treatable in DOT. Furthermore, a purported advantage of TAG over Tree-Insertion Grammars (Schabes and Waters, 1993) is that TAG can have 'wrapping adjunction'. Schabes & Waters (1993) give *deduce from* as an example, as in:

(10) John deduces that Mary invited Bob from smelling smoke.

However, DOP again would permit fragments containing nothing other than *deduce* and *from* (among other relevant fragments), allowing the link between these lexical items to be maintained. In any case, it is disputable whether wrapping adjunction is needed to model most languages. Dutch cross-serial dependencies would be an exception, and DOP1 cannot handle these examples.

2 LFG-DOT

We now propose the use of LFG-DOP (Bod & Kaplan, 1998) as the basis for an innovative MT system. We present here a number of possible translation models:

1. The first model (LFG-DOT1) performs constraint-based translation at the level of f-structure, using LFG's τ-equations to relate corresponding fragments in the two languages. This model takes advantage of monolingual LFG-DOP corpora to add 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.
2. The second model (LFG-DOT2) combines τ-equations with the γ-function, which links equivalent subtrees in a bilingual treebank.
3. The third model (LFG-DOT3) jettisons τ-equations and relies exclusively on γ to express the translation relation, as Way (2001) shows that the τ mapping cannot always produce the desired translation.
4. As may be expected, like DOT2, LFG-DOT3 suffers from limited compositionality. Therefore, we advocate the use of a restricted form of *Discard* in an 'extended transfer' phase to generalize the translation relation in a desirable manner, resulting in our final model, LFG-DOT4.

We shall see that this latter model describes the translation relation exactly as required, and furthermore overcomes the problems of LFG-MT and DOT models of translation.

2.1 LFG-DOT1: Translation via τ

This is a simple, linear model. Given a source language LFG-DOP treebank, the model builds a target f-structure f' from a source c-structure c and f-structure f, the mapping between them LFG-DOP-ϕ, and the LFG translation equations τ. From this target f-structure f', a target string is generated via a target language LFG-DOP model[3], as in (11):

(11)
$$\begin{tikzcd} c \arrow[r, "\text{LFG-DOP-}\phi"] & f \arrow[d, "\tau"] \\ c' & f' \arrow[l, "\text{LFG-DOP-}\phi'"] \end{tikzcd}$$

[3]Note that LFG-DOP-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.

The different components needed are a source language LFG-DOP model, the τ mapping, and a target language LFG-DOP model. The probability model for LFG-DOT1 in (12) is adapted from the probability model of Chang & Su (1993) for a speech-speech translation by omitting all terms pertaining to acoustic signals, rendering it suitable as a model for a text-based translation system:

(12) $$P(W_t \mid W_s) = \sum_{R_{s,t}} P(R_s \mid W_s).P(R_t \mid R_s, W_s).P(W_t \mid R_t, R_s, W_s)$$

That is, the conditional probability of the target string W_t given the source string W_s, $P(W_t \mid W_s)$, is equal to the probability of the representation R_s of the source string given that string, multiplied by the probability of the target representation R_t given the prior probabilities of the source representation and string, multiplied by the probability of the target string W_t given the prior probabilities of the source and target representations and the source string.

The probability model in (12) needs to be adapted to try to avoid the problem of sparse data—in our case, LFG-DOT fragments. We assume, therefore, a first order Markov assumption to approximate these probabilities, as in (13):

(13) $$P(W_t \mid W_s) \simeq \sum_{R_{s,t}} P(R_s \mid W_s).P(R_t \mid R_s).P(W_t \mid R_t)$$

That is, comparing (13) with (12), the assumption made is that the probability of the target string W_t is dependent solely on the prior probability of the target representation R_t, and not on the prior probability of the joint events R_t, R_s, W_s. The other simplification, however, does not incorporate Markov assumptions—it can be made given the nature of the linguistic representations we are using in our LFG-DOT models. Note that a full LFG-DOP $\langle c$, LFG-DOP-$\phi, f\rangle$ representation R_s *includes* the string W_s, as this occurs in the c-structure itself. In this case, therefore, $P(R_t \mid R_s, W_s)$ is always equal to $P(R_t \mid R_s)$. While this simplification is valid, the first approximation is of course wrong: the probability of the target string *is* influenced by the probability of the source string, but we choose to omit this information so as to simplify and simultaneously improve our translation model.

Returning to (13), for similar reasons as just outlined, one might wonder whether $P(W_t \mid R_t)$ may be omissable since it is trivially equal to 1, since W_t is always included in R_t. We shall make use of this observation in our probability models for LFG-DOT2, but we cannot do so for LFG-DOT1. The output from the transfer phase is a target f-structure, from which many possible strings may be generated (cf. free word order languages, for instance).[4] In addition to outputting a target f-structure, transfer in LFG-DOT2 also includes the production of the target c-structure tree via γ (i.e. the DOT2 model of translation), a constituent part of which is obviously the target string itself. In LFG-DOT2, therefore, we omit the $P(W_t \mid R_t)$ term (cf. (22) and following models), but this must be maintained in our probability models for LFG-DOT1.

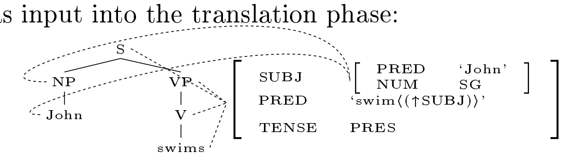
Determining the W_t that maximizes $P(W_t \mid W_s)$ involves making the following calculations if the probability model in (13) is used:

- $P(R_s \mid W_s)$: the source language LFG-DOP model, which can be calculated by the LFG-DOP model (Bod & Kaplan, 1998), and then by normalizing. That is, $P(R_s \mid W_s) = \frac{P(Rs,Ws)}{\sum\limits_{R'_s} P(R'_s,W_s)}$.
- $P(R_t \mid R_s)$: transfer, in terms of τ-equations, which can be estimated by $\frac{P(R_t,R_s)}{P(R_s)}$, i.e. dividing the probability of the f-structures R_t and R_s together, by the probability of R_s.
- $P(W_t \mid R_t)$: the target language LFG-DOP generation model, where $P(W_t \mid R_t) = \frac{P(W_t,R_t)}{\sum\limits_{W'_t} P(R_t,W'_t)}$

In order to see more precisely how an LFG-DOT Model 1 might work, let us take the simple example sentences in (14):

(14) a. John swims $\Longleftrightarrow$ Jan zwemt.
b. Peter laughs $\Longleftrightarrow$ Piet lacht.

LFG-DOT1 presumes a monolingual source treebank: if we assume English to be the source language, then four sentences can be analysed from the LFG-DOP fragments in the treebank built from the two English strings in (14). For (14a), LFG-DOP would produce the structures in (15) as input into the translation phase:

(15) 

We now need to consult the τ-equations in (16):[5]

---

[4] Note that a target language LFG-DOP model is able to compute the most likely string among all possible strings compatible with any such f-structure.

[5] Monolingual features are omitted for clarity.

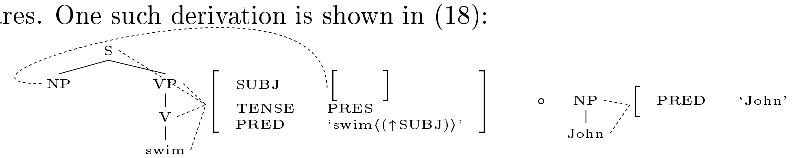
(16) a. *swims*: $(\tau\!\uparrow \text{PRED}) = \text{zwemmen}$, $\tau(\uparrow \text{SUBJ}) = (\tau\!\uparrow \text{SUBJ})$
b. *John*: $(\tau\!\uparrow \text{PRED}) = \text{‘Jan’}$

Taking the entry for *swims* as an example, the τ-equations stipulate that its translation is *zwemmen*, and that the translation of its SUBJ is the SUBJ of *zwemmen*. These τ-equations enable the target f-structure in (17) to be built:

(17)
$$\left[\begin{array}{ll} \text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & \text{‘Jan’} \\ \text{NUM} & \text{SG} \end{array}\right] \\ \text{PRED} & \text{‘zwemmen}\langle(\uparrow\text{SUBJ})\rangle\text{’} \\ \text{TENSE} & \text{PRES} \end{array}\right]$$

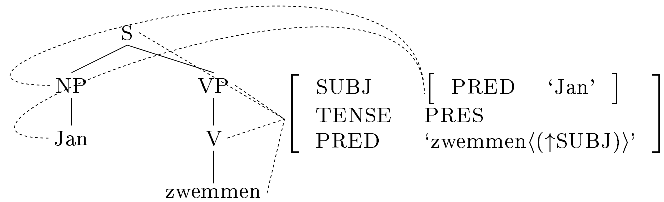
The target string *Jan zwemt* can now be generated from the f-structure in (17) via LFG-DOP-ϕ' in a target language LFG-DOP model.

The main advantage of LFG-DOT1 compared to LFG-MT is added robustness. LFG-DOT1 contains two monolingual LFG-DOP language models, so *Discard* can be run on both source and target sides. This means that LFG-DOT1 can cope with ill-formed or previously unseen input which LFG-MT would not be able to handle at all. Suppose that *John swim* is encountered as the source string, as in (2). In LFG-DOP, this can only be interpreted if *Discard* relaxes certain constraints in the f-structures. One such derivation is shown in (18):

(18)


That is, the `SUBJ:NUM:PL` path will have been relaxed in the sentential f-structure, with the `NUM = SG` feature removed from the rightmost NP f-structure. The NP c-structure can be substituted at the NP node in the leftmost c-structure, and the f-structures unified. The resultant underspecified 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 (18) 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 (19):

(19)



The 'translation' of the ill-formed string *John swim* would therefore be *Jan zwemmen*.

With respect to more complex examples, the advantage of this model over DOT1 is the availability of the explicit τ-equations to link source-target correspondences. Recall that DOT1 was unable to handle the *like* $\longleftrightarrow$ *plaire* case correctly in (1). LFG-DOT1 uses τ to express the translation relation, so the LFG-MT solution (20) to this relation-changing case can be availed of quite straightforwardly:

(20) *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*.

2.1.1 Summary

LFG-DOT1 can solve some of the problems of DOT1, and has certain advantages over DOT2. *Discard* improves the robustness of LFG-MT. However, in using the τ mapping for transfer, like LFG-MT it does not get some of the 'hard' translation cases right. Way (2001) shows that DOT2 can handle some of these cases correctly, including relation-changing (1) and headswitching cases (3)–(9). It seems sensible, therefore, to introduce the γ relation into our translation models in order to link source and target subtree fragments. This is the LFG-DOT2 model of translation.

2.2 LFG-DOT2: Translation via τ and γ

This model requires integrated bilingual LFG-DOP corpora, where each node n in a source c-structure tree c is related both to its corresponding f-structure fragment f (via LFG-DOP-ϕ) and its corresponding c-structure node n' in a target c-structure tree c' (via γ). In addition, each f-structure fragment s in a source f-structure f is related to its corresponding language fragment s' in a target f-structure f', via τ, as in (21):

(21)
$$\begin{tikzcd} c \arrow[r, "\text{LFG-DOP-}\phi"] \arrow[d, "\gamma"'] & f \arrow[d, "\tau"] \\ c' & f' \arrow[l, "\text{LFG-DOP-}\phi'"] \end{tikzcd}$$

Model 2 contains explicit links between both surface constituents and f-structure units in both languages, whereas LFG-DOT1 relates languages just at the level of f-structure (via τ). Consequently, LFG-DOT2 necessitates a source language LFG-DOP model, the γ mapping, a target language LFG-DOP model, and a probabilistic transfer component. The probability model for LFG-DOT2 uses the general LFG-DOP probability model (Bod & Kaplan, 1998) to compute probabilities for the full $\langle c, f \rangle$ representations, and not for f-structures alone (as in LFG-DOT1). The LFG-DOT1 probability model (13) can be reduced to (22):

(22) $$P(W_t \mid W_s) = \sum_{R_{s,t}} P(R_s \mid W_s).P(R_t \mid R_s)$$

$P(W_t \mid R_t)$ can be omitted since it is trivially equal to 1, since if R_t is given, then the target string W_t can always be seen, this being a component part of the target c-structure. Given this, it is clear that $P(R_s \mid W_s) = P(R_s)$, so (22) is further reducible to (23):

(23) $$P(W_t \mid W_s) = \sum_{R_{s,t}} P(R_s).P(R_t \mid R_s)$$

An alternative, still simpler formulation of (23) is (24):

(24) $$P(W_t \mid W_s) = \sum_{R_{s,t}} P(W_t, R_{s,t} \mid W_s)$$

(24) is further reducible to (25) since W_t and W_s are in R_t and R_s respectively as $R_{s,t}$ consist of the full $\langle c, f \rangle$ representation pairs for the source and target strings W_s and W_t:

(25) $$\sum_{R_{s,t}} P(R_{s,t})$$

As usual in LFG-DOP, $P(R_s)$ and $P(R_t)$ are equal to the sum of the derivations which make up their representations, where the derivations are equal to the product of all linked LFG-DOT fragments which can be combined to form these derivations, and the probability of each linked LFG-DOT fragment is calculated as its frequency relative to all linked fragments in the corpus as a whole. Note that none of these LFG-DOT models which employ the full $\langle c,$ LFG-DOP-$\phi, f \rangle$ representation for strings require any additional Markov assumptions. The basic translation units are pairs of linked LFG-DOP fragments, and the basic stochastic event is the combination of two linked LFG-DOP fragment pairs.

If we translate *John swims* $\Longleftrightarrow$ *Jan zwemt* in LFG-DOT2 (no *Discard*), the source structures input into the translation phase are those in (15). An LFG-DOT2 treebank[6] contains source (as in (15)) and target ϕ-links, as well as γ-links between translationally equivalent parts of the source and target c-structures. Adding the LFG-DOT2 treebank for (14b) (which would mirror that for *John swims* $\Longleftrightarrow$ *Jan zwemt* except for the leaves on the c-structure trees and f-structure PRED values) enables us to handle the two new translations in (26):

(26) a. John laughs $\longleftrightarrow$ Jan lacht.
b. Peter swims $\longleftrightarrow$ Piet zwemt.

As with the DOT examples, the probabilities of all four translations are the same as those assigned under DOP. The fact that we have to factor in f-structure fragments makes no difference here given the trivial corpus: source c-structures are combined in an identical fashion to Tree-DOP, and as in DOT, the target c-structures are built simultaneously. Given that source and target fragments are $\langle c,\ \text{LFG-DOP-}\phi, f\rangle$ triples, the source and target f-structures are built up in a corresponding synchronous fashion. Depending on which LFG-DOP probability model is chosen, the process by which the structures are built differs: whether we use just the Tree-DOP Root category matching condition (model M1), LFG's Uniqueness condition (M2), Coherence check (M3) or Completeness checks *post hoc* will affect neither the actual structures built nor their probabilities for the simple translation pairs studied here.

The main reason for the addition of the γ operation in LFG-DOT2 is that the correct translation is not always achievable via τ-equations. Way (2001) has shown that DOT2 is able to produce correct translations for a range of phenomena, so harnessing γ to the τ-equations in a probabilistic transfer component should provide better results. Maintaining an f to f' translation engine in addition to γ increases the likelihood of achieving the correct translation—even if this is not proposed as the most probable translation via τ, given that this function will only ever produce very few translation candidates, we can guarantee in almost all cases that it is 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.

Alternatively, if the γ and τ translation relations are permitted to operate on the same probability space, then the applicability of τ needs

[6] Treebanks of this type would serve as input into LFG-DOT Models 2-4. LFG-DOT1 treebanks consist of separate source and target c-structure trees and their corresponding f-structures. Target f-structures are created by means of τ-equations.

to be restricted in terms of its impact on the probabilities of candidate translations. We use Good-Turing to limit the effect of *Discard* on the probabilities (cf. Way, 1999; Bod, 2000), and propose the application of the same technique here. In this way the two functions (γ and τ) are combined, but τ is limited to a smaller section of the probability space. Therefore the γ links have priority over the τ-equations, but they work in harness. Whichever approach is taken, the production of multiple instances of the same translations via different functions can be viewed as mutual confirmation of the best translation.

Using LFG-DOP as the source and target language models overcomes the shortcomings of both Tree-DOP and LFG. LFG is considerably more sophisticated than most published DOP language models. DOP models are tree-based which restricts their applicability as general purpose language models. In contrast, LFG handles (almost all) non-surface phenomena with ease.

Using τ-equations ensures that the correct translation will be produced in almost all cases. We have seen that DOT1 cannot ensure this. Using τ-equations in isolation, as in LFG-MT, necessitates the ranking of a number of output f-structures by a human expert. Unlike LFG-DOT1, LFG-DOT2 enables automatic ranking, and pruning (if required) of f-structures. Furthermore, LFG-DOT2 is more robust than LFG-MT, in that *Discard* can produce generalized fragments which may be able to deal with input outside the scope of the 'ungeneralized' LFG-DOT2 database. In these cases, LFG-MT has no option but to offer no translation candidates at all.

The DOT2 γ function copes properly with translational phenomena which cause problems for the LFG τ-equations. LFG-DOT2 maintains the τ translation relation to increase the chances of the correct translation being produced. We discussed how the two functions might best co-operate, with the γ relation taking priority. Nevertheless, one has to ask the question as to whether it is fruitful to try and integrate the two translation relations. We need the presence of f-structure information in order to allow *Discard* to run and thereby make LFG-DOT more robust than LFG-MT. However, *Discard* can operate whether the f-structures are linked via τ or not, so it would appear that the τ operation itself is not needed. In some cases, τ-equations provide extra evidence in favour of the correct translations, where owing to the restricted nature of γ-links, only semi-compositional linked fragments are available. Indeed, this flaw influences the changes made in our final translation model, LFG-DOT4.

In the next two models the τ operation is omitted, with the translation relation stated solely in terms of γ.[7]

2.3 LFG-DOT3: Translation via γ with Monolingual Filtering

The LFG-DOT3 model proposed in this section contains the DOT2 links between source and target c-structures, but with additional syntactic functional constraints which prevent ungrammatical structures such as (2) from being formed (except via *Discard*, in much the same way as in (18) 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. The LFG-DOT3 architecture is shown in (27):

(27)
$$\begin{tikzcd} c \arrow[r, "\text{LFG-DOP-}\phi"] \arrow[d, "\gamma"'] & f \\ c' \arrow[r, "\text{LFG-DOP-}\phi'"'] & f' \end{tikzcd}$$

The probability model given in (23) for LFG-DOT2 can also be used for LFG-DOT models 3 and 4. That is, the probability of a target string W_t given a source string W_s can be calculated by multiplying the probability of the source representation R_s by the probability of the target representation given the prior probability of the source representation. The only difference between models 2 and 3 (or 4) concerns the nature of the linguistic representations involved: in LFG-DOT2, the languages are linked via the DOT2 γ link as well as via τ, whereas only γ is used as an interface in subsequent models.

If we adopt (23) as the probability model for LFG-DOT3 (and 4), we require the following components:

- $P(R_s)$: the probability of the source structures, a full $\langle c$, LFG-DOP-$\phi, f\rangle$ representation;
- $P(R_t \mid R_s)$: transfer, in terms of the DOT2 γ function;
- $P(R_t)$: the probability of the target structures, a full $\langle c$, LFG-DOP-$\phi, f\rangle$ representation.

[7] We leave for future work the question as to whether this approach is fruitful for languages which differ significantly at the level of surface structure, e.g. English and Warlpiri. In such cases, perhaps an LFG-DOT1 or LFG-DOT2 model may be better to relate translational equivalents at the level of f-structure rather than c-structure.

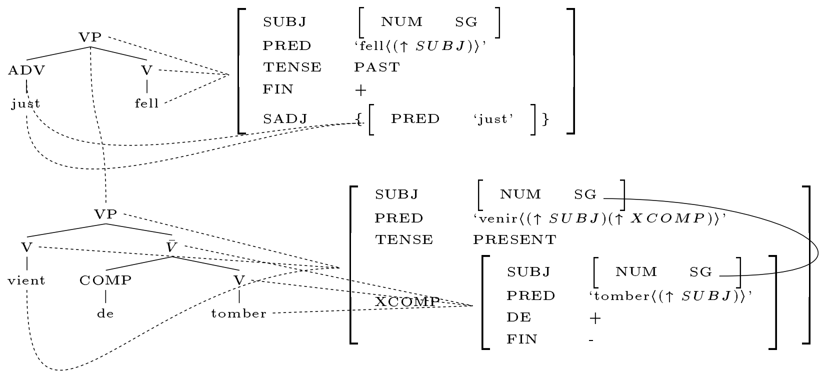


FIGURE 3 The *just* $\longleftrightarrow$ *venir de* headswitching case in LFG-DOT3

The probability of the source representation $P(R_s)$ is calculated in the normal way (cf. (13) and subsequent discussion). The transfer component $P(R_t \mid R_s)$ can be estimated by $\frac{P(R_t, R_s)}{P(R_s)}$, i.e. dividing the probability of the linked source and target $\langle c, f \rangle$ structures R_t and R_s, by the probability of the source representations R_s. Comparing (23) with (13), note again that a full LFG-DOP source representation contains the source string, so $P(R_s \mid W_s)$ is omissable without loss of information.[8]

As an example of how the LFG-DOT3 model of translation works, let us consider the *just* $\longleftrightarrow$ *venir de* headswitching case. In terms of LFG-DOT3, the translation relation is shown in Figure 3. The γ link between semantically equivalent elements in the source and target c-structures can be seen on the VP nodes. As in DOT2, *fell* is not considered 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 (cf. discussion in section 1.1 regarding translational equivalence). Hence this translation fragment can only be reused by substituting this pair with associated singular NP subjects at the appropriate nodes in an S-linked fragment. In this respect, as with DOT2 (and LFG-DOT2), this LFG-DOT3 model continues to suffer from limited compositionality. We address this concern further in the next section which deals with the

[8]The only difference between the probability models for LFG-DOT3 and LFG-DOT4 is that a small amount of the probability space in LFG-DOT4 is given over to lemmatized translation pairs in the extended transfer phase. Way (2001) proposes that this be kept to a minimum via Good-Turing: unless some limit is placed on newly formed fragments, the *Discard*-generated fragments quickly take over from non-*Discard* fragments to the detriment of the overall probabilities of sentences. Bod (2000) demonstrates this effect for monolingual treebanks; in the context of translation models, this will lead to correct translations being less preferred than 'ungrammatical' variants.

LFG-DOT4 model.

2.3.1 Summary

LFG-DOT1 expresses the translation relation by means of the τ mapping, whereas LFG-DOT2 combines τ with the γ function. LFG-DOT3, meanwhile, eschews the τ relation and relies completely on γ to relate translationally equivalent source and target fragments. The τ relation was abandoned for several reasons:

- it is unable to always express the correct translation relation;
- the f-structures produced may not be acceptable to the target grammar (an example of the 'subset problem', cf. Way, 2001:25f.);
- the γ relation is capable of producing correct translations where this cannot be guaranteed with τ.

Way (2001) illustrates how LFG-DOT3 improves on LFG-MT as well as pure tree-based models. He also proves that LFG-DOT3 and LFG-DOT2 are different models by attempting to reconstruct the τ-mapping from other constraints, showing that while this is possible in cases of simple transfer, in examples containing more complex translational phenomena, the τ-equations are not inferable.

The presence of functional information in LFG-DOT3 prevents ill-formed structures such as (2) from being formed, unless *Discard* is used to process such linked pairs in dealing with ill-formed input. LFG-DOT models, therefore, have a notion of grammaticality which is missing from DOT2. Importantly, this can be used to guide the probability models in the manner required. Bod (2000) shows that unless *Discard*-generated fragments are restricted to a fixed, small portion of the probability space, wrong analyses may be preferred over correct ones. Way (2001) illustrates the same problem with respect to translational phenomena.

In using γ to express the translation relation, we observed that LFG-DOT3 is open to the same criticism as DOT2 in that both models suffer from limited compositionality. Given this, the minimal translation relations cannot be stated. Therefore, an LFG-DOT3 database will need to be extremely large in order for such fragments to have even a small probability of participating in the combinatory process with other fragments. LFG-DOT4 does not suffer from this drawback, rendering the likelihood of its fragments' usefulness as translationally relevant examples significantly higher.

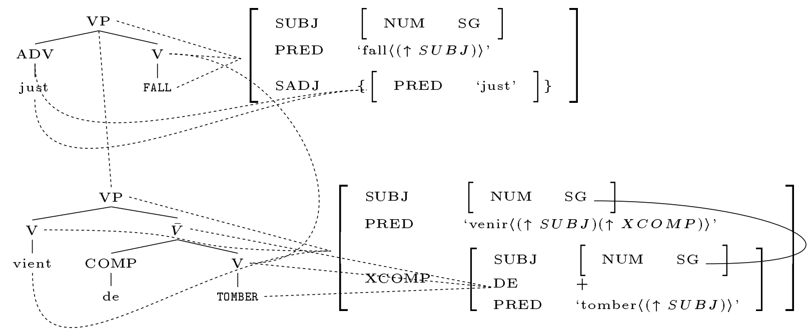
2.4 LFG-DOT4: Translation via γ and 'Extended Transfer'

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 3, we would like to be able to 'relax' some of the constraints in order to map $\langle fell, tomber\rangle$ to make these linked fragments more general, and hence more useful. In so doing, we would remove this problem of limited compositionality.

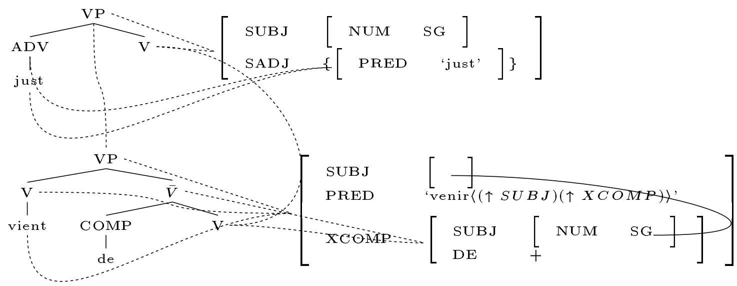
In LFG-DOT4, the basic translation relation is expressed by γ, as in LFG-DOT3. In LFG-DOT4, however, there is a second application of *Discard*, by which 'lemmatized' forms are arrived at on which 'extended transfer' can be performed. *Discard* relaxes constraints in order to produce a set of generalized 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 3, they can be regarded as translationally equivalent. Given this, $\langle fell, tomber\rangle$ are linked and lemmatized, as in (28):

(28)



Now that ⟨`FALL`,`TOMBER`⟩ are linked, they can be deleted to produce the generalized form of the translation relation, namely (29):

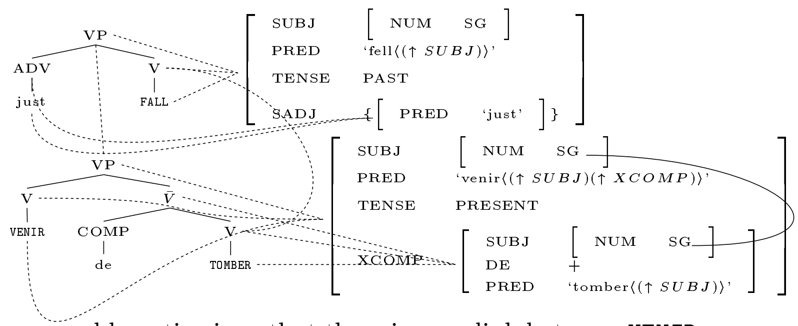
(29)



If fragment pairs such as (29) 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. Bod, 2000), such translations will receive *some* probability, whereas the semi-compositional variants from which they were derived may not be able to produce *any* translation.

The examples in (28) and (29) are, however, a little incomplete. The reader will observe that we have been rather selective in our choice of lemmatization. Given that the lemmatized forms in these examples are peripheral to the translation relation, we have chosen simply to apply *Discard* to these elements so that they may be deleted. In practice, if *all* verbs in Figure 3 are subjected to the lemmatization process, we will instead end up with a different linguistic object, namely (30):

(30)



This remains unproblematic given that there is no γ link between `VENIR` and a node in the English tree. Therefore, we are still able to link ⟨`TOMBER`, `FALL`⟩ and end up with the desired, generalized translation relation ⟨*just*, `VENIR` *de*⟩.[9]

Rather than relying on a post-editing phase to transform cases such as *L'homme* `VENIR` *de* `TOMBER` $\longleftrightarrow$ *The man just* `FALL`, one wonders whether the target grammar might be able to 'fill in' any missing features

[9] We retain lemmatized forms in the 'translations' produced for post-editing. Any other format more suitable to effective post-editing may, of course, be chosen.

and thereby enable a correct, target string to be formed. This is what target grammars have to do in any case, namely coerce target structures in generation: the objects input to the generation phase necessarily have to be augmented by rules in the target grammar so as to produce a *bona fide* target representation from which a well-formed target string can be read off.

What we want to do is allow the target grammar to fill in features (but *not* overwrite them) only in f-structures produced by *Discard*. With respect to the target structures in (30), for instance, the only elements which need to be fleshed out are the V nodes in the c-structure. In a treebank of any size, we can hypothesize that there will be many other f-structures which are subsumed by such an f-structure, several of which will contain features for PRED and TENSE. These will be linked to their respective target c-structures, some of which we can expect to have the correct surface realization of these features, namely *vient* and *tomber*. If these lexemes occur with a reasonable frequency (to be arrived at empirically), then a final processing stage can be foreseen whereby these word forms replace the templatized forms in the output target c-structure. We leave this as an experiment for future work.

2.4.1 Summary

Most translation examples will be handled in LFG-DOT4 in exactly the same way as in LFG-DOT3. The outstanding problem with LFG-DOT3, however, is its retention of the DOT2 problem of limited compositionality. LFG-DOT4, like LFG-DOT3, expresses the basic translation relation by means of the γ function. LFG-DOT4 adds an 'extended transfer' step to LFG-DOT3 by producing lemmatized forms using a second application of *Discard*. This extension overcomes the problem of limited compositionality.

Lemmatization (via *Discard*) barely affects the probability of previous, correct LFG-DOT translations which were arrived at in a limited compositionality manner. We propose the use of Good-Turing to limit the scope of any newly created translations via lemmatization by restricting the amount of probability space available to the extended transfer phase. How much of this probability space should be given over to the lemmatized translation pairs needs to be established empirically via extensive testing when appropriate corpora become available. The work of Frank *et al.* (2001) on semi-automatic derivation of LFG corpora from treebank resources would appear promising in this regard. Despite the fact that lemmatized translations will obtain only small probabilities, the very fact that they are obtained and offered up as translation candidates is what is important: for some of the more complex cases, almost the exact translation example needed to be found in DOT2 and LFG-DOT3 for a translation candidate to be obtained. Extended transfer generalizes translation fragments considerably, thereby increasing the likelihood that such fragments will indeed play a role in the derivation of new translations.

Finally, we hypothesized a further stage of processing in which grammatical target structures might be formed from those generated via *Discard* in the extended transfer phase of translation, by reference to other similar structures in the treebank. In this way, correct morphological forms may replace lemmatized forms so that correct target strings are generated, thereby obviating the need for post-editing of such output.

3 Conclusions

We have presented 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 on DOT1, which is not guaranteed to produce the correct translation when this is non-compositional and considerably less probable than the default, compositional alternative. DOT1's 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. As soon as this deviates to any significant degree between languages, DOT1 has a significant bias in favour of the incorrect translation.

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.

DOT2 addresses the failings of DOT1 by redefining the composition operation and providing an improved probabilistic model. It appears that 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. 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. We discussed how the two functions might best co-operate, with the γ relation taking priority. Ultimately, given that the τ-equations fail to derive the correct translation in all cases, we omit 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. Importantly, this can be used to guide the probability models in the manner required. However, both 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.

Stating the translation relation solely between ⟨*source, target*⟩ trees, as in LFG-DOT3 and LFG-DOT4, works so successfully as we are freed from the restriction of having to relate local trees (cf. Way, 2001:221). LFG-DOT4, like the other LFG-DOT models, is a robust system. One needs, however, to ensure that the structures obtained via *Discard* are of use, especially on the target side. These structures subsume the structures required for the successful generation of a well-formed target string. We described how lemmatized forms occurring in target c-structures may be replaced by the appropriate surface form by comparing the partial f-structure information to which they are linked to other, complete f-structures in the treebank. Nevertheless, we need to ensure that those derivations using fragments produced by *Discard* are restricted to a small, fixed part of the probability space in our models in order to ensure a preference for grammatical structures. We propose the adoption of the Good-Turing method to cope with unknown words, to limit the scope of *Discard*-generated fragments, and to limit the amount of the probability space available to lemmatized translation pairs.

References

[1] 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.

[2] Bod, R. (2000): 'An Empirical Evaluation of LFG-DOP', in COLING: *Proceedings of the 19th International Conference on Computational Linguistics*, Saarbrücken, Nancy and Luxembourg.

[3] 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.

[4] Chang, J. and K. Su (1993): 'A Corpus-Based Statistics-Oriented Transfer and Generation Model for Machine Translation', in *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in MT*, Kyoto, Japan.

[5] Frank, A., J. van Genabith and A. Way (2001): 'Treebank vs. X-BAR based Automatic F-Structure Annotation', in *Proceedings of 6th International Conference on Lexical Functional Grammar (LFG-2001)*, Hong Kong.

[6] 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.

[7] 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.

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

[9] Poutsma, A. (2000): *Data-Oriented Translation: Using the Data-Oriented Parsing framework for Machine Translation*, MSc thesis, University of Amsterdam, The Netherlands.

[10] 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.

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

[12] Schabes, Y. and R. Waters (1993): 'Stochastic Lexicalised Context-Free Grammar', in *Proceedings of the 3rd International Workshop on Parsing Technologies*, Tilburg, The Netherlands, pp.257–266.

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

[14] Way, A. (1999): 'A Hybrid Architecture for Robust MT using LFG-DOP', *Journal of Experimental and Theoretical Artificial Intelligence* **11**:441–471, Taylor and Francis, London.

[15] Way, A. (2001): *LFG-DOT: A Hybrid Architecture for Robust MT*, PhD Thesis, University of Essex, UK.

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