LFG-DOT: Combining Constraint-Based and Empirical Methodologies for Robust MT

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

Email: away@compapp.dcu.ie

Abstract

The Data-Oriented Parsing Model (DOP, [1]; [2]) has been presented as a promising paradigm for NLP. It has also been used as a basis for Machine Translation (MT) — Data-Oriented Translation (DOT, [9]). Lexical Functional Grammar (LFG, [5]) has also been used for MT ([6]). LFG has recently been allied to DOP to produce a new LFG-DOP model ([3]) which improves the robustness of LFG. We summarize the DOT model of translation as well as the DOP model on which it is based. We demonstrate that DOT is not guaranteed to produce the correct translation, despite provably deriving the most probable translation. Finally, we propose a novel hybrid model for MT based on LFG-DOP which promises to improve upon DOT, as well as the pure LFG-based translation model.

1 Introduction

Neither of the main paradigmatic approaches to MT, namely rule-based and statistical, currently suffice to the standard required. Nevertheless, each contains elements which if properly harnessed should lead to an overall improvement in translation performance. It is in this new hybrid spirit that our search for a better solution to the problems of MT can be

Proceedings of NODALIDA 1999, pages 243-254

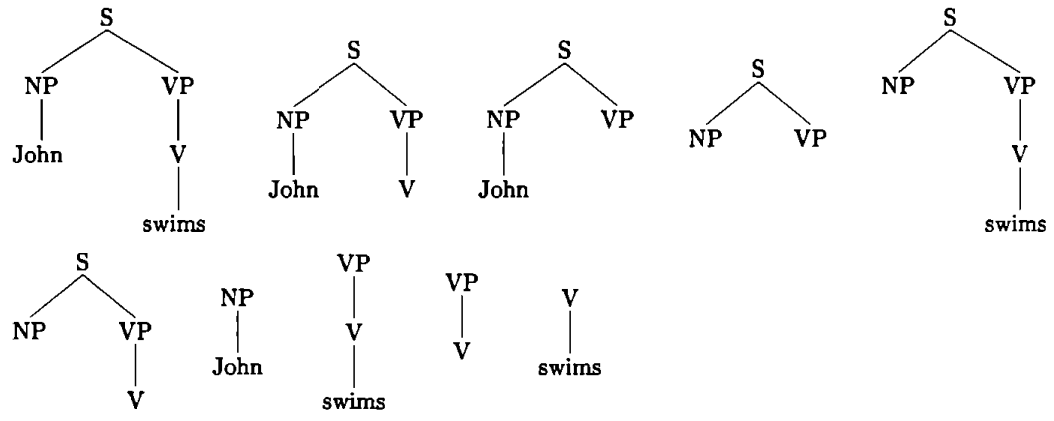
seen. We propose that combining DOP ([1];[2]) with the conventional transfer rules of LFG ([6]) may derive a new model for MT, LFG-DOT, which promises to improve upon DOT, as well as the pure LFG-based translation model.

2 The DOP Architecture for NLP

DOP language models ([1];[2]) assume that past experiences of language are significant in both perception and production. DOP prefers performance models over competence grammars, in that abstract grammar rules are eschewed in favour of models based on large collections of previously occurring fragments of language. New language fragments are processed with reference to already existing fragments from the corpus, which are combined using probabilistic techniques to determine the most likely analysis for the new fragment.

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

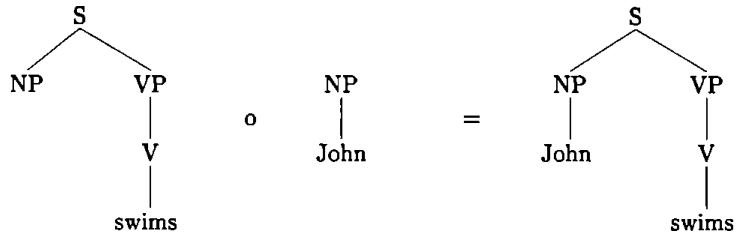
(1)



The full set of DOP trees derived from the sentence *John swims* are those in (1).

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

(2)



Finally, the chosen **probability model** for Tree-DOP is based quite simply on the relative frequencies of fragments in the corpus.

These elements enable representations of new strings to be constructed from previously occurring fragments in a number of ways. If each derivation t has a probability P(t) (i.e. its relative frequency), then the probability of deriving a Tree-DOP representation is the sum of the probabilities of the individual derivations, as in (3):

(3)

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

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

(4)

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

Given these formulae, the probability of the derivation for *John swims* in (2) is $\frac{1}{6}$. This is calculated by multiplying together the probability of each of the two tree fragments involved in the derivation, namely those in (5):

(5) P(t = [**NP vp[v[swims]]**] | root(t) = S).P(t = **[np[John]]**| root(t) = NP) = $\frac{1}{6} \cdot \frac{1}{1} = \frac{1}{6}$

The probability of the *parse* of *John swims*, however, is calculated by summing all derivations resulting in the parse-tree for the sentence (as (3) shows), which, given the trivial corpus in

(1), is 1. However, adding the fragments from a new sentence *Peter laughs* to the treebank in (1) allows us now to derive the probability of two new strings – *Peter swims* and *John laughs* – with respect to this small corpus of tree fragments. In this way, it can be seen that DOP handles unseen data on the basis of previous experience – despite the fact that we have never seen either new sentence before, we are able to process them compositionally, on the basis of previously occurring fragments of each in our corpus. Each tree which can play a part in combining together with other trees to form a representation for a sentence is used to contribute to the overall probability of that representation given the corpus.

2.1 Opportunities for Hybridity—LFG DOP

DOP-based approaches are necessarily limited to those contextual dependencies actually occurring in the corpus, which is a reflection of surface phenomena only. Given its facility to capture and provide representations of linguistic phenomena other than those occurring at surface structure, the functional structures of LFG have been allied to the techniques of DOP to create a new model, LFG-DOP ([3]), which adds a measure of robustness not available to models based solely on LFG. We suggest that this framework has the potential to be utilised for MT.

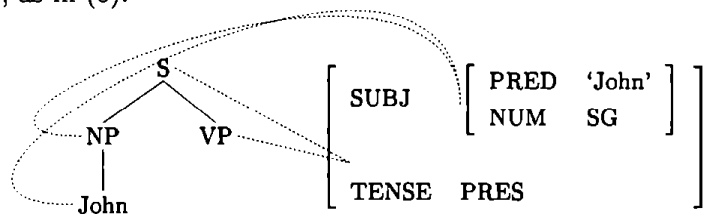
As with DOP, LFG-DOP needs to be defined using four parameters. Its **representations** are simply lifted *en bloc* from LFG theory, so that each string is annotated with a c-structure, an f-structure, and a mapping ϕ between them, with well-formedness conditions operating solely on f-structure, as usual.

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

Frontier operates as in DOP, selecting a set of nodes in the newly created subtree, excluding the root, and deleting all subtrees dominated by these selected nodes. Furthermore, it deletes all ϕ-links of these erased nodes together with any semantic form corresponding to the same

nodes, as in (6):

(6)



which illustrates the ability of *Root* nodes to access certain features (TENSE, here) even after subnodes have been deleted. (6) can be pruned still further by applying a third, and new operation, *Discard*, to the TENSE feature. *Discard* adds considerably to LFG's robustness by providing generalised fragments from those derived via *Root* and *Frontier*.

Composition is also a two-step operation. C-structures are combined by left-most substitution, as in DOP, subject to the matching of their nodes. F-structures corresponding to these nodes are then recursively unified, and the resulting f-structures are subjected to the grammaticality checks of LFG.

Finally, $P(f \mid CS)$ denotes the probability of choosing a fragment f from a competition set CS of competing fragments. [3] describe four possible competition sets linked to the **probability models** for LFG-DOP: (i) a straightforward extension of the DOP probability model where the choice of a fragment depends only on its *Root* node and not on the Grammaticality conditions of LFG; (ii) c-structure nodes must match, and f-structures must be unifiable if two LFG fragments are to be combined, i.e. taking into account the LFG Uniqueness condition as well as the *Root* category; (iii) furthermore, the LFG Coherence check is enforced at each step; and (iv) finally, all LFG grammaticality checks, as well as the DOP category-matching stipulation, are left to the end. Note that in models (i)–(iii) the category matching condition is enforced on-line whilst all LFG checks are either performed on-line or *post hoc*, whereas given the non-monotonic nature of the Completeness check, this can only ever be enforced *post hoc*.

3 Data-Oriented Translation (DOT)

[9] has developed a DOP-based model of translation — Data-Oriented Translation — which relates POS-fragments between two languages (English and Dutch), with an accompanying probability. Once a derivation for the source language sentence has been arrived at, the target structure is assembled, and a string produced. Since there are typically many different

derivations for the source sentence, there may be as many different translations available. As is the case when DOP is used monolingually, the probability of a translation is calculated by summing the probabilities of all possible derivations forming the translation. Poutsma shows that the most probable translation can be computed using Monte-Carlo disambiguation, and exemplifies this using sentence idioms, where corresponding source-target translations are linked at all possible nodes.

3.1 Some Limitations of DOT

DOT is an interesting model, yet it fails to capture the correct translation when this is non-compositional and considerably less probable than the default, compositional alternative. When LFG-DOP MT (LFG-DOT) is used instead this problem may be overcome. Furthermore, DOP's statistical model also gives a "level of correctness" figure to alternative translations. This is useful in cases where the default translation in LFG-MT (and in many other systems) cannot be suppressed when the specific translation is required. For example, assuming the basic default rules in (7):

(7) a. commettre ⇔ commit

b. suicide ⇔ suicide

in order to deal with the sentences in (8):

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

b. Le suicide est tragique ⇔ Suicide is tragic

we would get the wrong translation where *John commits suicide* ⇔ **John commet le suicide* (cf. *John se suicide*). We would like specific rules to override the default translation where applicable, but this is not possible in LFG-MT, so we would get both translations here, i.e. a correct one (via the specific τ-equations in (10)) and a wrong one (via the default τ-equations, required to translate *commettre* as *commit* in other circumstances). Assuming a DOP treebank built from the French sentences in (8) as well as *Marie se suicide*, the ill-formed string *Jean commet le suicide* is preferred (in the French language model) about half as much again as the correct alternative *Jean se suicide*. There are several reasons for this: the preference for *Jean* as subject of *commettre*, the co-occurrence of *le* and *suicide*, plus the fact that *commettre* is followed by an NP consisting of a Det + N sequence. Note

also that these results are obtained with the same number of instances of each verb — in a larger corpus *commettre* would surely greatly outnumber instances of *se suicider*.

This is by no means an unexpected result. As an example, in the *LOB Corpus*, there are 66 instances of *commit* as a verb (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 thousands of instances of *commettre*. Given occurrences of *suicide* as an NP in French corpora, it is not an unreasonable hypothesis to expect that the wrong, compositional translations will be much more probable than those derived via the specific rule.

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

4 LFG-DOT: A New Theory of Translation

The DOT model cannot explicitly relate parts of the source language structure to the corresponding, correct parts in the target structure. One line of investigation which we now develop that can overcome this linear restriction is to use LFG-DOP ([3]) as the basis for an innovative MT system, using LFG's τ-equations to relate translation fragments between languages.

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

Using separate language corpora, this simple, linear model builds a target f-structure f' from a source c-structure c and f-structure f, the mapping between them ϕ, and the tau-equations τ. From this target f-structure f', a target string is generated via the standard LFG generation algorithms ([7]; [11]). The probability of the target f-structure R_t being the translation of the source string W_s is:

$$(9) \qquad \begin{aligned} \max_{R_t} P(R_t \mid W_s) &= \max_{R_t} \sum_{R_{t,s}} P(R_s \mid W_s).P(R_t \mid R_s, W_s) \\ &= \max_{R_t} \sum_{R_{t,s}} P(R_s \mid W_s).P(R_t \mid R_s) \end{aligned}$$

incorporating a Markov assumption that the target f-structure's derivation from a source string (via ϕ and τ) is independent of the original words involved: it is dependent *solely* on the monolingual LFG-DOP representation assigned. This is an attempt to avoid as much as possible the sparse data problem, given that in all probability we will never have enough LFG-DOP fragments to model these numbers with any reasonable accuracy. The components needed given (9), therefore, are (i) a source language LFG-DOP model, $P(R_s \mid W_s)$; (ii) the τ mapping (the translation model) plus the associated probabilities that a source f-structure produces a target equivalent, $P(R_t \mid R_s)$.

The advantage of this model over DOT is the availability of the explicit τ-equations to link source-target correspondences, as in (10):

(10) *commit*: ($\tau\uparrow$ PRED) = se suicider, τ(↑ SUBJ) = ($\tau\uparrow$ SUBJ), (↑ OBJ PRED) $=_c$ suicide

Using LFG τ-equations ensures the derivation of the correct target f-structure, along with some wrong alternatives (here) via the default rules. We cannot be sure that the generation of a target string via the correct target f-structure will be a more probable translation than any wrong alternative, but it will exist as one of a small number of high-ranking candidate solutions from which the final translation can be selected. Of course, we may instead choose to derive the target string using a target language LFG-DOP model (via ϕ') rather than the standard LFG generation algorithms, in which case the probability model in (9) needs to be adapted to incorporate $P(W_t \mid R_t)$, where again we presume that the target string generation is independent of all source language representations: it is dependent solely on the τ-equations derived from the source f-structure.

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

Here we have integrated language corpora, where for each node in a tree c, we relate it both to its corresponding f-structure fragment f and its corresponding target c-structure node c', and for each source f-structure fragment, we relate that to its target language fragment in f-structure f', via τ. The probability model used this time is:

$$(11) \qquad \overset{Max}{t} P(t \mid s) = \overset{Max}{t} \sum_{R_{t,s}} P(t, R_{t,s} \mid s) = \overset{Max}{t} \sum_{R_{t,s}} P(R_t \mid s)$$

where now $R_{t,s}$ are the full $\langle c, f \rangle$ representation pairings for the target and source strings, respectively. Our basic units are pairs of linked LFG-DOP fragments (cf. the linked DOP fragments in DOT, [9]), and the basic stochastic event is the combination of two linked LFG-DOP fragment pairs. Thus, we compute the probability of $P(t \mid s)$ by the sum of the probabilities of all R_t, R_s pairs that generate t and s (and, ultimately, of course, choosing that t for which this probability sum is maximal), where the probability of an R_t, R_s pair is computed as the sum of the probabilities of its derivation-pairs; each derivation-pair is the product of its linked fragment-pairs; and each linked fragment-pair has a probability equal to its normalized relative frequency. Bod & Kaplan discuss four different ways of calculating the probability of an (unlinked) fragment, depending on which LFG grammaticality checks (if any) are integrated into the competition sets assumed (cf. section 2.1).

The principal reason for hypothesising the γ function in this model is that it is reasonable to assume, as [9] has shown, that valuable information concerning the final formulation of the target string can be influenced by the source c-structure. In this way we have two pieces of information at hand with which to build the target string—the γ and ϕ' functions, which if they can be properly harnessed, should bring about a better translation, given the extra evidence that is being brought to bear in its generation.

4.3 Semi-Automatic Creation of LFG & LFG-DOP Corpora

A major problem for researchers interested in LFG and LFG-DOP is the absence of suitable, extensive corpora. Given this, in order to demonstrate practically the feasibility of LFG-DOT, we have begun to develop our own LFG and LFG-DOP corpora ([10]).

Initially we took the publicly available set of 100 sentences of the *AP Treebank* ([8]). Despite its small size, this was sufficiently large to demonstrate the plausibility of our approach. One

particular entry is:

(12)
```
A001  39 v
[N The_AT march_NN1 N][V was_VBDZ [J peaceful_JJ J]V] ._.
```

We then automatically extract the rules from this corpus (following the method of [4]), and create automatically LFG-macros for each lexical category:

(13)
```
macro(at(Word),FStr) :-
  FStr:spec === Word.

macro(jj(Word),FStr) :-
  FStr:pred === Word.

macro(nn1(Word),FStr) :-
  FStr:pred === Word,
  FStr:num  === sg.

macro(vbdz(_Word),FStr) :-
  FStr:tense  === past,
  FStr:pred   === be.
```

We then annotate the extracted rules with LFG functional schemata by hand:

(14)
```
rule(n(A), [at(B),nn1(C)]) :-
   A === B,
   A === C.

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

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

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

and 'reparse' the original treebank entries, *not* the strings, simply by recursively following the tree annotations provided by the original annotators. In so doing the interpreter solves the constraint equations associated with the grammar rules and lexical macros involved in the parse, returning single f-structures, as in:

(15)
```
subj : spec : the
       pred : march
       num  : sg
xcomp : pred : peaceful
        subj : spec : the
               pred : march
               num  : sg
tense : past
pred  : be
```

In order to produce target f-structures, all that is necessary is to add τ-equations to the

lexical and structural rules, and reparse the treebank entries. Once these target f-structures exist, we can test out the translation models and report results.

5 Conclusions

The DOT translation system, despite provably deriving the most probable translation, is not guaranteed to produce the best, or even a correct translation, since it is unable to explicitly link exactly those fragments which are playing the decisive role in translation.

[3] have shown how DOP and LFG can be integrated to provide a powerful mechanism for the treatment of parsing. We described how such a model may be extended to provide a robust solution for the problems of MT in the spirit of the current trend for hybrid approaches. LFG-DOT promises to improve on previous attempts at LFG-MT, particular where robustness is concerned, being able to handle both unseen and ill-formed input with relative ease. It also ensures that the correct target f-structure is input into the generation process. It is reasonable to expect LFG-DOT to outperform pure statistics-based systems, in having the additional facility of grammatical information at hand to use where necessary.

Much of this work is ongoing, and a number of issues remain for the future, especially the automatic creation of large LFG-DOP corpora necessary as training and test data for the translation models. This will complete the development of the systems described, leading to greater experimentation on a larger scale.

References

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

[2] Bod, R. (1998): *Beyond Grammar: An Experience-Based Theory of Language*, CSLI Publications, Stanford, California.

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

[4] Charniak, E. (1996): "Tree-bank grammars", in AAAI-96, *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, MIT Press, pp.1031-1036.

[5] Kaplan, R. & 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.

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

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

[8] Leech, G. and R. Garside (1991): "Running a Grammar Factory: on the Compilation of Parsed Corpora, or 'Treebanks' ", in: S. Johansson and A.-B. Stenström (eds), *English Computer Corpora: Selected Papers and Research Guide*, Mouton de Gruyter, Berlin, pp.15–32.

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

[10] Van Genabith, J., A. Way and L. Sadler (1999): "Semi-Automatic Generation of F-Structures from Treebanks", in *Proceedings of LFG-99*, Manchester, UK.

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