Optimality Theory Style Constraint Ranking in Large-scale LFG Grammars

Anette Frank, Tracy Holloway King, Jonas Kuhn, John Maxwell

XRCE, Xerox PARC, IMS Stuttgart, Xerox PARC

Proceedings of the LFG98 Conference

University of Queensland, Brisbane

Miriam Butt and Tracy Holloway King (Editors)

1998
CSLI Publications
http://www-csli.stanford.edu/publications/

1 Introduction

Ambiguity is one of the key problems in the construction of scalable natural language processing systems. Even though the use of packed representations facilitates efficient parsing with ambiguity, the disambiguation problem is almost certain to show up when dealing with the output of an application.

Many ambiguities that realistic systems with broad-coverage grammars are faced with include readings that appear very implausible for a particular sentence, i.e., with the given lexical material and context. However, looking more closely at the structure of the sentence will often reveal that putting syntactic constraints on the constructions involved to generally exclude the unwanted reading would not be justified, because with different lexical material or in a different context this reading may well be the most plausible one, or may even be the only possible one. It is desirable to be able to express a dispreference for infrequent constructions, without having to rule them out in all cases.

Here we propose a straightforward extension of the LFG projection architecture that incorporates ideas from the theoretical literature on Optimality Theory (OT; see Prince and Smolensky (1993); Bresnan (1996a,b); i.a.). On top of the classical constraint system of existing LFG grammars (Butt et al. (to appear)) a new projection, o-structure, determines a preference ranking on the set of analyses for a given input sentence: a relative ranking is specified for the constraints that appear in the o-projection, and this ranking serves to determine the winner among the competing candidates. The Optimality Theoretic constraints are overlaid on the existing grammar and hence do not fundamentally alter the basic tenets of LFG theory.

As experience with large grammars for English, French, and German shows,[1] this OT mechanism can be very effective in filtering syntactic ambiguity. Even though in certain cases the preference constraints stated in the grammar are faced with exceptions or counterexamples, they can give valuable results in domain specific applications and allow a selective and focused approach to particular ambiguities during grammar development. If in addition to the syntactic constraints, other sources of "soft" information are available, such as selectional restrictions on the semantic classes of verb arguments, the OT-style ranking provides a framework for having such information sources interact.

A further advantage of the addition of OT to the LFG architecture is that the robustness of a grammar can be increased by adding low-ranked fallback rules. Such rules can allow for the parsing of common grammatical mistakes (e.g., subject-verb agreement mistakes) and marginal constructions (e.g., misplaced adverbials). This potential for increased robustness is invaluable in real world applications of LFG in which the material to be parsed is in a less ideal format than the average linguistic example sentence. Finally, using the same grammar in parsing and generation can be facilitated by applying the OT-style preference mechanism: while a grammar must accept a wide variety of alternative syntactic structures in parsing, generation should be restricted to a subset of 'preferred' construction alternatives.

Section 2 introduces the implementation of Optimality Theory constraints and the place of OT in the projection architecture of LFG. Section 3 is the main body of the paper and discusses possible applications for OT theory in large scale LFG grammars. Finally, section 4 provides discussion and possible avenues for further development.

[1] Currently, the grammars are being implemented in the Xerox Linguistic Environment (XLE) as part of the Parallel Grammar (ParGram) Project (Xerox Research Centre Europe, IMS Stuttgart, Xerox PARC).

2 Formal implementation

This section discusses a particular implementation of OT in an LFG system. Section 2.1 introduces the idea of an optimality projection as the basis for an OT-style preference mechanism. The basic approach is exemplified by looking at a syntactic ambiguity found with PPs. Section 2.2 shows how the constraints encoded in the optimality projection are introduced in the grammar and provides more details about the formal interpretation that determines the relative ranking of analyses.

2.1 Optimality projection

For each candidate analysis of a given input (an input string, in the parsing task), a record is kept of the rules and constraints used. For this record, an extra level of representation is assumed: o-structure, which is formally a multiset of constants ("optimality marks"), projected from c-structure. Optimality marks are explicitly introduced by o-descriptions within the grammar (see (2) for an example where optimality marks are used within a disjunction at the level of functional annotations). The o-structure of a phrase is the multi-union of the o-structure of its constituents plus any local optimality marks.

In the specification of the constraint ranking for a grammar, the optimality marks, as used in the o-descriptions, are ordered, marking the satisfaction of certain constraints (i) as positive (and for some even more positive than for other positive constraints), or (ii) as negative (and accordingly for some more negative than for others). In (1), MARK1 is the most positive and MARK4 the most negative. All constraints without an explicit marking are treated as NEUTRAL. (As will be seen in the next section, the marks below NEUTRAL are in fact divided into three subfields.)

(1) OPTIMALITYRANKING   MARK1 MARK2 NEUTRAL MARK3 MARK4.

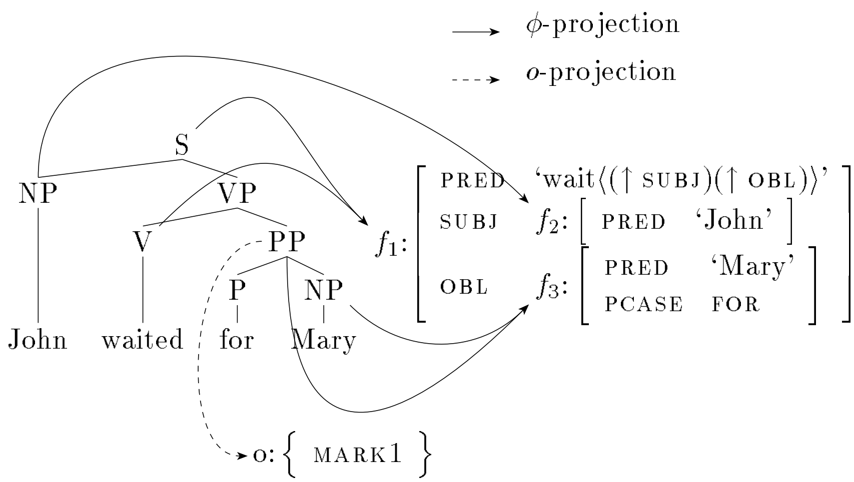
(2) $$\text{VP} \rightarrow \text{V} \left(\begin{array}{c} \text{NP} \\ (\uparrow \text{OBJ}) = \downarrow \end{array} \right) \quad \begin{array}{c} \text{PP}* \\ \left\{ \begin{array}{c} (\uparrow \text{OBL}\ ) = \downarrow \\ \text{MARK1} \in o* \\ \\ \downarrow \in (\uparrow \text{ADJUNCTS}) \\ \text{MARK2} \in o* \end{array} \right\} \end{array}$$

Applying the ordinary algorithm (parsing and feature constraint resolution) to an input string produces the set of candidates that enter the competition under the given constraint ranking. To determine the winner, just the o-structures of the candidates have to be considered: the winning structure(s) will be the one(s) containing the fewest instances of the most negative mark. If this does not produce a unique candidate, the second most negative mark is counted, and so on. If all the negative marks fail to single out a candidate, for the remaining structures the positive marks are considered successively, starting from the most positive mark. Here the candidates with the *greatest* number of instances win.

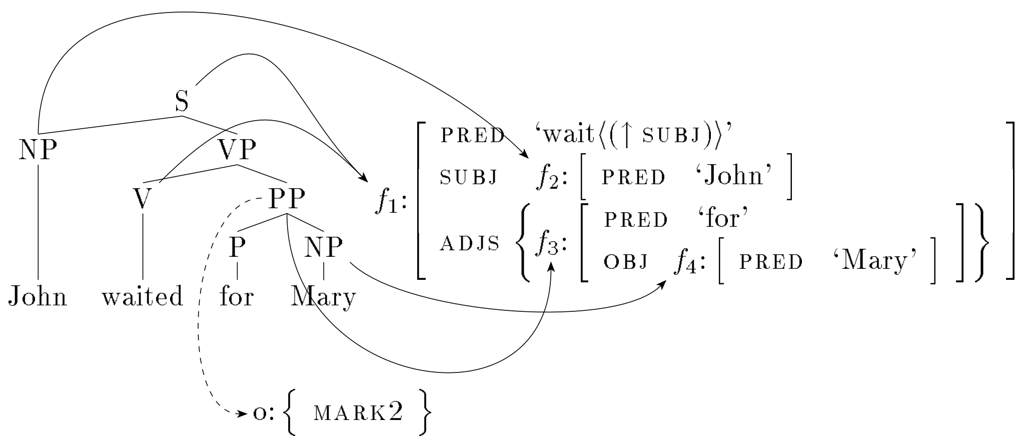
Assume, for example, that the disjunction under PP in rule (2) is the only source of optimality marks for a certain sentence (as in the competing analyses (4) and (5) for sentence (3)), and that the constraint ranking is as in (1). Candidate (4) wins over (5), since neither analysis introduces any negative optimality marks, but (4) contains an instance of the highest positive mark, MARK1, while (5) does not.

(3) John waited for Mary.

(4)



(5)



However, the bottom disjunct under PP in (2) which introduces the PP's f-structure as an element of the VP's ADJUNCTS set will not generally be suppressed. With a verb not introducing an OBL function, like *sleep* in *John slept for three hours*, the analysis corresponding to (4) will be ruled out by LFG's Coherence condition (which is not violable), so the (5)-type analysis will have no competitor and will, trivially, win.

We can also use this PP example to show the limitations of a constraint ranking exclusively based on syntactic information: sentence (6) will give rise to exactly the same ambiguity as (3); however this time the different choice of lexical items reveals that structure (5) is the intended one.

(6) John waited for three hours.

Adding information about certain semantic classes to the system of ranked constraints can solve some (though certainly not all) problems of this type: if temporal expressions like *three hours* or *a day* can be identified, a new mark MARK0 can be introduced in the adjunct analysis of a *for*-PP containing a temporal expression. Ranking MARK0 higher than MARK1 would cause the original syntactic preference of argument PPs to be overruled: the (5)-type analysis with the adjunct PP would have an o-structure set {MARK0, MARK2} and would beat (4), which just has {MARK1}. Another way to go would be to express ranked constraints on the semantic class of the verb arguments (cf. sec. 3.2).

In principle, the proposed OT-style constraint ranking provides a framework for such a combination of soft constraints from different sources. It should be noted however that there will always be cases where on the basis of the linguistic information provided to the system, an incorrect alternative wins.[2] Nevertheless, even a mainly syntax-based constraint ranking turns out to be useful when writing large-scale grammars. We come back to this discussion in section 3.2.

2.2 Interpretation of ranked constraints

The most common metric used to compare analyses in the OT literature is to compare the marks on the two analyses from most dispreferred to least dispreferred until there is a mark that has a different number of instances in the two analyses. When there is a mark where the two analyses differ, then the one that has the fewest instances of that mark is chosen. The implementation discussed here extends Optimality Theory by allowing preference marks. If the dispreference marks do not distinguish between two analyses, then the preference marks are compared from most preferred to least preferred. When there is a mark where the two analyses differ, then the analysis that has the greater number of instances is chosen.

The ranked constraints introduced in the grammar are ranked from highest to lowest.

(7) MARK1 NEUTRAL MARK2 UNGRAMMATICAL MARK3 NOGOOD MARK4

NEUTRAL, UNGRAMMATICAL, and NOGOOD are special marks. An analysis with a NEUTRAL mark is equivalent to one having no mark at all.[3] Its only purpose is to separate preference marks from dispreference marks. Marks listed before NEUTRAL are preference marks. Marks listed after NEUTRAL are dispreference marks. Note that in traditional approaches to Optimality Theory, marks are listed from most dispreferred to least dispreferred; however, XLE lists them in the other order because of the preference marks.[4] The four types of marks (preference, dispreference, ungrammatical, and nogood) are discussed below, each one with an brief example of how it might be used in a grammar.

Preference Marks Preference marks are used when one out of two, or more, readings is preferred. For example, preference marks can be used to state a preference for the multiword analysis of technical terms: in general, when the multiword expression reading is possible, it is the preferred one. Typical English examples are seen in (8), while (9) shows a translation triple in which a noun-noun compound multiword expression is used in English, a morphological compound in German, and a N-P-N multiword expression in French.

(8) a. Xerox DocuColor 70

b. fast forward

(9) a. print quality

[2] For example, modifying (6) to (i) would probably defeat the ranking scheme just outlined.

(i) John waited for a clear day.

[3] By default, marks that appear in the grammar but not in the ranking are treated as being NEUTRAL.

[4] To facilitate the modification of the ranking without having to edit the grammar and lexicon files, it is also possible to collect some marks into an equivalence class by enclosing them in brackets. A declaration of the form:

OPTIMALITYRANKING (MARK1 MARK1a) (MARK2 NEUTRAL) MARK3 UNGRAMMATICAL
MARK4 NOGOOD MARK5

is interpreted in such a way that MARK1a and MARK1 count as preference marks of identical strength and MARK2 is treated as equivalent to NEUTRAL, i.e., it is effectively ignored.

b. Druckqualität
print-quality

c. qualité d' impression
quality of printing

By marking the lexical entries of multiword expressions with a preference optimality mark, the analysis involving the multiword expression will be preferred over the analytic analysis whenever both analyses are possible, as in (10a). However, if there is no valid analysis for the multiword expression, as in (10b), the analysis that uses the individual word entry/ies will still be possible.

(10) a. The [print quality] of this printer is good.

b. I want to [print] [quality] documents.

Dispreference Marks Dispreference marks (those below NEUTRAL and above UNGRAMMATICAL) are generally used on rare constructions that are grammatical and as such are parsed by the grammar, but are less likely to occur. The dispreference mark ensures that the construction surfaces only when no other, more common, analysis is possible. Consider the case of NPs headed by adjectives. A simple example of a valid use of this construction is seen in the German example (11).

(11) Meistens kauft die größere Firma die kleinere.
Mostly buys the larger company the smaller
'Usually, the larger company buys the smaller one.'

However, having a version of the NP rule which allows an NP to consist of an adjective with an optional determiner permits the grammar to try to build such NPs in many places where they are implausible. An example of such a situation is seen in (12), which would have two readings if there were no OT markings constraining the use of adjective headed NPs.

(12) Nachts fallen helle Farben auf.
at night stand bright colors out

a. $[_{\mathrm{NP(nom)}}$ helle Farben]
reading = 'At night, bright colors stand out.'

b. $[_{\mathrm{NP(nom)}}$ helle] $[_{\mathrm{NP(dat)}}$ Farben]
reading = 'At night, the colors notice bright ones/bright ones strike colors.'

The reading in (12a) in which *helle Farben* forms an NP is the desired one. However, the reading in (12b) in which *helle* forms an NP headed by an adjective and *Farben* a canonical NP is also possible (the verb *auffallen* can either be intransitive or take a dative object). To constrain this extra, infelicitous reading, a dispreference mark can be introduced in the part of the NP rule which derives the adjective headed nouns, as shown in the rule fragment (13).

(13) a. OPTIMALITY RANKING ... NEUTRAL AHEADNP UNGRAMMATICAL NOGOOD.

b. NP ⟶ { ...

(DET)	A
(↑SPEC)=↓	↑=↓
	AHEADNP ∈ $o*$

... }

In general, it can be difficult to decide whether to use a preference or dispreference mark in order to express a preference for one of two analyses. There are two issues here: the possible interaction between the marks and other, unrelated analyses, and what happens when one of the analyses is missing. If you add a preference mark to the preferred analysis, then it will be preferred over all unmarked analyses, whether they are related or not. On the other hand, if you add a dispreference mark to the dispreferred analysis, then it will be dispreferred even when the preferred analysis is not present.

For instance, recall our example of PP ambiguities from Section 2.1, where we wanted to express a preference for obliques over PP adjuncts in order to handle the syntactic bias for obliques that shows up in sentences like:

(14) John waited for Mary.

There are two ways to do this: to add a preference mark to obliques or to add a dispreference mark to PP adjuncts. If we add a preference mark to obliques, as we have done above, then the oblique analysis will be preferred over all unmarked readings, even those that are completely unrelated to these two. For instance, suppose that the parser was given the following ambiguous input from a speech recognizer:

(15) John waited {for | four} {hours | ours }.

This has the following readings, among others:

(16) a. John waited [$_{\text{PPobl}}$ for ours].

b. John waited [$_{\text{PPadj}}$ for ours].

c. John waited [$_{\text{PPobl}}$ for hours].

d. John waited [$_{\text{PPadj}}$ for hours].

e. John waited [$_{\text{NPadj}}$ four hours].

Adding a preference mark to obliques would cause the first and third readings to be preferred. If we had a more sophisticated preference scheme, then we could avoid the preference mark on the third reading since it is a time expression. However, we would still have the unintended result that the first reading would be preferred over the fifth reading, which is otherwise unrelated.

On the other hand, if we disprefer PP adjunct readings, then we get an unintended dispreference when the preferred reading is missing, such as in (17).

(17) a. [$_{\text{NP}}$ Fruit flies] [$_{\text{V}}$ like] [$_{\text{NP}}$ a banana]

b. [$_{\text{NP}}$ Fruit] [$_{\text{V}}$ flies] [$_{\text{PP}}$ like a banana] (dispreferred)

In (17a), *fruit flies* is treated as a compound noun subject, *like* as a transitive verb, and *a banana* as the direct object. Since there is no PP, no dispreference mark is incurred. In (17b), *fruit* is the subject, *flies* the verb, and *like a banana* is an adjunct PP. As such, (17b) incurs a dispreference mark and will not surface even though in this case it is at least as preferred a reading as that in (17a).

In this case, we would probably be better off using a preference mark on the obliques, since intuitively there seems to be a preference for obliques rather than a dispreference for PP adjuncts. This shows preference for obliques can even override pragmatic considerations, as in (18).

(18) He painted the wall with cracks.

Here, many people prefer the reading which states that the result of the painting is that cracks appear on the wall, even though this is pragmatically dispreferred.

Another consideration when deciding between preference and dispreference marks is which analysis is easier to mark. For instance, it is much easier to mark a multi-word expression with a preference mark than it is to mark its components with a dispreference mark that only comes into play when *all* of the components are used in a particular way. However, this causes multi-word expressions to be preferred even when compared to unrelated analyses, as in (19) with the multi-word expression *a bat out of hell.*

(19) Fruit flies like a bat out of hell.

Whether this is a good or bad thing remains to be determined.

Ungrammatical Marks It is also possible to have ungrammatical marks. These are used to mark error rules which parse ungrammatical constructions (this is useful for building robust grammars, see below). A simple example of this is relaxing the subject-verb agreement constraint. A basic form of this is shown in (20) for the English third singular verbal ending *-s*. In cases where subject-verb agreement is observed, the first disjunct delivers an analysis where no OT mark is used. However, when subject-verb agreement is violated, only the second disjunct can be chosen, which introduces an ungrammatical mark NoSVAgr. This structure will only surface if there is no grammatical analysis. Similar constraints could be added for other person and number combinations. If the only solutions are ungrammatical, then XLE marks this by adding an asterisk before the number of solutions whenever the number of solutions is reported.

(20) a. OPTIMALITYRANKING NEUTRAL UNGRAMMATICAL NoSVAgr NOGOOD.

b. -s { (↑SUBJ NUM)=SG
(↑SUBJ PERS)=3
| NoSVAgr ∈ *oM*∗ }

The difference between dispreference marks and ungrammatical marks is that analyses that are marked ungrammatical are ignored unless there are no grammatical analyses, either preferred or dispreferred. In applications, this is only relevant when analyses are being passed to a disambiguation module. If there are grammatical analyses, then none of the ungrammatical analyses will be passed to the disambiguation module. If there are no grammatical analyses, then the ungrammatical analyses are passed to the disambiguation module, along with their relative rankings.

NOGOOD Marks The NOGOOD marks indicate that the analysis is always bad, even if there is no other analysis. The purpose of this is to allow fine-grained filtering of the grammar. For instance, a grammar might be annotated with TRACTOR-MANUAL and VERBMOBIL-CORPUS marks that indicate constructions that are only used in those domains. If these marks are listed after NOGOOD, then these constructions are treated as being inconsistent. As such, if one of these marks occurs conjunctively in a rule or lexical item, then the rule or lexical item will still appear in the output, but the mark will appear as an inconsistent constraint in the display and no further structure will be built on top of it. For example, a special rule to parse section headers might be written for a corpus which contains many such headers, but this rule might be undesirable for other uses of the grammar. A similar example is seen in (21) in which a pre-sentential number is permitted as a sentence tag number in the Verbmobil corpus (ranking (21a)),[5] but could be excluded elsewhere (ranking (21b)).

[5]In the ranking in (21a) it might be desirable to bracket the VERBMOBIL-CORPUS mark with the *neutral* mark since these structures are not necessarily preferred, but simply permitted.

(21) S ⟶ (NUMBER) NP VP
(↑TAG)=↓ (↑SUBJ)=↓ ↑=↓
VERBMOBIL-CORPUS ∈ $o*$

a. OPTIMALITYRANKING VERBMOBIL-CORPUS NEUTRAL NOGOOD.

b. OPTIMALITYRANKING NEUTRAL NOGOOD VERBMOBIL-CORPUS.

3 Applications in large-scale grammars

In this section, we discuss in more detail how the basic mechanism just introduced can be successfully exploited within large classical LFG grammars, exemplified by implemented grammars for English, French, and German. Apart from the alluded-to application for a flexible filtering of syntactic ambiguity, low-ranked constraints can be used to mark special rules devised either for marginal input (e.g., that often found in spontaneous speech, misspellings, etc.), or for constructions not (yet) covered systematically (e.g., guessing that a word with a capital first letter is a proper name). These two ideas increase the robustness of the grammar. Finally, an additional promising application of constraint ranking is the parametrization of a single grammar for use in parsing vs. generation by specifying two different rankings for these two processes. These areas for application of OT preference constraints are discussed in the subsequent sections.

3.1 Spurious Ambiguity

Besides the filtering of syntactic ambiguities (as discussed above for the example of PP ambiguities), it is tempting to utilize OT to constrain spurious ambiguity as well. However, as discussed in this section, spurious ambiguity is almost always best constrained by other means.

A prime example of spurious ambiguity appears with coordination. As the data in (22) illustrate, the c-structure rules for NPs have to accomodate a number of different levels of coordination. With such rules and no ranking, the analysis of coordinated plural NPs like *cats and dogs* will contain spurious ambiguity. In this case, there would be the three analyses seen in (23).

(22) a. [the cousins] and [a sister]

b. a [brilliant singer] but [mediocre guitarist]

c. the numerous [friends] and [relatives] of Peter's

(23) a. N: [[[[cats] and [dogs]]]]

b. N': [[[[cats]] and [[dogs]]]]

c. NP: [[[[cats]]] and [[[dogs]]]]

(24) OPTIMALITYRANKING LEVEL1 LEVEL2 LEVEL3 NEUTRAL.

$$\mathrm{N} \rightarrow \underset{\textsc{Level1} \in o*}{\mathrm{N} \quad \mathrm{CONJ} \quad \mathrm{N}}$$

$$\mathrm{N}' \rightarrow \left\{ \begin{array}{ccc} \mathrm{AP}* & \mathrm{N} & \\ \mathrm{N}' & \mathrm{CONJ} & \mathrm{N}' \\ & \textsc{Level2} \in o* & \end{array} \right\}$$

$$\mathrm{NP} \rightarrow \left\{ \begin{array}{ccc} (\mathrm{DET}) & \mathrm{N}' & \\ \mathrm{NP} & \mathrm{CONJ} & \mathrm{NP} \\ & \textsc{Level3} \in o* & \end{array} \right\}$$

Introducing subsequently ranked optimality marks at the different levels of coordination (as in the small NP grammar in (24) which prefers structurally low coordination) will at first glance appear to solve the problem. However, if this is done, then a coordinated NP such as that in (25) is no longer ambiguous even though this is an instance of *true* ambiguity.

(25) a. [old [men and women]]

b. [[old men] and women]

The OT grammar in (24) will only produce (25a) since the OT mark LEVEL1 prefers that structure over the other one. However, in this case both parses are possible and only context can determine which one is preferred. Filtering out just the spurious cases of ambiguity is not possible with OT theory in this implementation. Instead, the instances of spurious ambiguity can be constrained by more judicious writing of the c-structure rules, such as ruling out N' → N when N is coordinated.

Some cases of spurious ambiguity can, however, be successfully filtered by OT constraints. For example, since French has rather fixed word order, the VP rule is not binary, but encodes the order of dependent constituents. To allow for the frequent alternation of object NP and PP order, however, there is a PP position preceding the object NP, besides the one that follows it. This leads to spurious ambiguities for verbs with a single oblique or dative object PP, like (26).

(26) Le livre appartient à Jean.
The book belongs to John.

Either PP-constituent position can derive the dative object. Annotation of the first PP-constituent with a dispreference mark PREOBJ-PP successfully filters the unwarranted ambiguity.[6]

[6] An alternative way of filtering this kind of ambiguity is to use intersection to restrict the derivation of the first PP position to expansions of VP where the NP object position is filled. However, this would mean that the VP rule has to be duplicated, causing redundancy.

3.2 Filtering of syntactic ambiguity

As mentioned, there is a limitation on the use of OT to filter syntactic ambiguity: for many kinds of ambiguities we will always find counterexamples, i.e., there will always be a case in which the OT constraints choose the incorrect reading. However, despite this drawback the inclusion of the OT constraint system is very useful, for reasons which are discussed here and in subsequent sections.

Before going into these, let us briefly sketch an example that to our knowledge does not suffer from the problem of counterexamples. In French, some verbs allow for optional complex predicate formation. The verb *laisser* in (27) is of this type: (27a) is a non-complex control construction, where the controlled NP object is placed between the main predicate and the embedded verb. (27b) is the complex construction, where the embedded verb's external argument surfaces as a postverbal NP object. Now, (27c), where the external argument of *venir* is a pronominal clitic, is structurally ambiguous: the sentence can be analyzed as involving a complex or non-complex construction. Since in all other cases the grammar will disambiguate the two construction possibilites, we can put a general preference or dispreference mark for either the complex or non-complex verb entry of ambiguous verbs like *laisser* that filters the ambiguity in cases like (27c), thus producing a single optimal analysis.[7]

(27) a. Jean laisse le conducteur venir.
John lets the driver come

b. Jean laisse venir le conducteur.
John lets come the driver

c. Jean le laisse venir.
John him lets come

However, to take up the much more frequent example of syntactic PP-ambiguities treated above, it is easy to find counterexamples to the general preference for obliques. We already mentioned a case where a general preference constraint for obliques as opposed to adjunct PPs gives the wrong result: *John waited for three hours.*

There are several possible conclusions to draw from such observations. One is to be aware of the problem of 'false guesses', but make use of these preference constraints nevertheless. A first, and simply pragmatic reason for this can be to help grammar writers in their daily task of grammar development, i.e., to prevent them from being overwhelmed with too many analyses. In addition to such pragmatic factors, experience has shown that if the application is limited to specific domains,[8] counterexamples are not very frequent. To a certain extent, then, certain types of preferences can be fine tuned for specific types of corpora. This is especially the case for lexical ambiguities, to be discussed below.

Occasional mistakes in preferences are also less harmful if the application incorporates an interactive disambiguation tool where first the most preferred (optimal) analyses are presented to the user, but on demand lower-ranked analyses can be accessed, stepwise, by exploiting the relative ranking determined for the total set of analyses. Even in a fully automatic setting, one might opt for different modes of application, allowing diverse strengths of reliance on the preference modules, and admit a limited number of additional lower-ranked analyses to be transferred to subsequent

[7]Although speakers perceive a very slight interpretation difference between the complex and non-complex construction, this difference does not show up, e.g., in translation. Thus, filtering this kind of ambiguitiy should do no harm.

[8]In the ParGram project grammar development mainly focuses on technical documentation.

processing modules. Given the large number of ambiguous analyses that can be produced by large-scale grammars, there is still considerable gain in such a relaxed model of application.

Ultimately, the most promising approach is to incorporate more external knowledge sources into the definition of preferences. In the case of PP ambiguities this could be selectional restrictions on verb arguments. For *wait* we can expect a lexicon that encodes selectional restrictions to assign the oblique argument slot a concept class that subsumes the concept class of *Mary*, but not that of *three hours*. A second "level" of preference constraints can thus be added to the existing system of preference constraints, which marks analyses as preferred if they are in accordance with the verbs' selectional restrictions. Below is a sketch of how this could be defined in the corresponding verb entries that contain oblique argument slots.

(28) wait: V $(\uparrow \textsc{pred})=$ 'wait⟨$(\uparrow \textsc{subj})(\uparrow \textsc{obl})$⟩'
$(\uparrow_r \textsc{arg1-cl}) = \{ \text{cl1 cl2} \}$
$(\uparrow_r \textsc{arg2-cl}) = \{ \text{cl1 cl5} \}$
$\{ ((\uparrow \textsc{obl})_r \textsc{ cl}) \ll (\uparrow_r \textsc{arg2-cl})$
$\mid ((\uparrow \textsc{obl})_r \textsc{ cl}) \not\ll (\uparrow_r \textsc{arg2-cl}) \quad \textsc{noselres} \in o* \}.$

Sentences where the concept class of the PP argument is not subsumed ($\not\ll$) by selectional restrictions of the corresponding verb argument slot are assigned a dispreference mark NOSELRES. In the case of *John waited for Mary*, where the selectional restrictions are satisfied, no dispreference mark is introduced. Due to the general preference mark for obliques, the oblique reading will be the most optimal one. In the example *John waited for three hours*, where the concept class of *three hours* is not subsumed by the selectional restriction classes for the oblique argument, NOSELRES will be added to the set of OT constraints. In this case, the oblique analysis is assigned two OT marks: NOSELRES and OBLIQUE (the preference mark for obliques), whereas the adjunct analysis is treated as neutral. Due to the dispreference mark NOSELRES, the oblique analysis will be correctly dispreferred.

This setup is, however, still not perfect. Consider the following example from French. In (29a) the oblique reading is correctly preferred, whereas (29b) is a case that is truly ambiguous. Even with the amendment taking into account selectional restrictions, the preference mechanism would still disprefer a correct analysis: in (29b) there is no violation of selectional restriction classes for the oblique argument, so OBLIQUE will be the only OT mark, yielding a preference of the oblique reading.

(29) a. Il renonce au voyage.
He abandons the journey

b. Il renonce au premier essai.
He abandons the first attempt
He abandons on the first attempt

A better solution, at least in this case, is therefore to drop the original general preference mark for oblique PPs (OBLIQUE) completely and retain the dispreference for oblique arguments in case the arguments' selectional restrictions are not met, as proposed in (28).

One has to be aware, though, that even a perfectly ranked combination of all possible linguistic sources of information would never completely compensate for the additional extra-linguistic information a human language user can easily exploit in the disambiguation task.

3.3 Lexical ambiguities

Another frequent source for ambiguities in large grammar systems is lexical ambiguity. The XLE implementation provides the LFG grammars with a direct interface to rich morphologies for English, French, and German. As a consequence, for many types of categories, especially nouns, there are no explicit LFG lexicon entries. Such entries are built on the fly by exploiting the information delivered by the morphological entries: a typical tree defined by so-called *sublexical rules* for nouns, interfacing to morphological entries, is depicted in (30a). (30b) shows the functional annotations of the morphological tags that define the corresponding functional features, while (30c) defines the stem as the value of the PRED feature. Together these rules define the f-structure in (30d) for the French "unknown" noun *utilisation*.

(30) a.

N			
N_STEM	GEND_TAG	NBR_TAG	N_TAG
utilisation	+FEM	+SG	+NOUN

b. +FEM GEND_TAG (↑GEND)= fem.
+SG NBR_TAG (↑NUM)= sg.

c. +MUNKNOWN N_STEM (↑PRED)= '%stem'.

d. [PRED 'utilisation'
GEND fem
NUM sg]

Since the morphologies are very rich, this provides a very powerful lexical device for the LFG grammars. The flip side of this is that it is a significant source of lexical ambiguity. For example, the French morphology has noun entries for many strings that are homonyms of function words: *est* (is/east), *si* (if/B (music)), *la* (the (fem)/A (music)), *a* (has/letter 'a'), *été* (been/summer), *être* (be/human being), etc. In many syntactic configurations it is possible to derive analyses that make use of these noun entries, in addition to those for the intended function words. An example is given in (31), where *est* can be analyzed as being the 3rd person form of the auxiliary verb *être* (be) or as the noun *est* (east) in a compound noun phrase *défaut est* (error east).[9]

(31) a. [$_S$ [$_{NP}$ Le défaut] [$_{VP}$ [$_{AUX}$ est] [$_V$ corrigé]]]
the error is corrected
'The error is corrected.'

b. [$_{NP}$ Le [$_{NMOD}$ défaut est] [$_{VPAP}$ corrigé]]
the error east corrected
'the corrected error east'

Optimality Theory can be used very efficiently to disprefer such alternative analyses by assigning these rare noun readings a dispreference mark RARE-NOUN. Especially if these noun readings are extremely unlikely to appear in the relevant text domain (e.g., *la* and *est* in technical documentation for copiers), this can be done without harm. Note that even with a dispreference mark on *est* as a noun, we still get the intended analysis for sentences like *Il faut s'orienter vers l'est – One must get*

[9]Note that for header phrases the grammar has to admit both noun phrases and sentences without punctuation.

one's bearings relative to the east. The presence of the determiner enforces the NP analysis, ruling out a verbal analysis. Since this is the only syntactic possibility, the NP analysis is "optimal" in this case despite the dispreference mark.

3.4 "Technical" usages

When using a grammar to parse a large corpus, OT can have a number of other uses which help to increase the robustness of the grammar and to minimize the amount of time spent hand altering the grammar and/or corpus.

In grammar engineering, constraint labelling and ranking can be used as a simple and flexible way of manipulating a grammar for diagnostic purposes. Kuhn and Rohrer (1997) use NOGOOD marks to create different grammar versions with switched off parts of rules; comparing the behavior of the manipulated grammar versions reveals sources of efficiency problems in the way the grammar is specified. Using labelled constraints has the advantage that across the grammar versions, just the OT ranking has to be modified, which helps to maintain controlled circumstances and allows the grammar writer to perform the same comparison at different stages of grammar development.

Kuhn, Eckle-Kohler and Rohrer (1998) employ ungrammatical marks in grammar-based semi-automatic lexicon acquisition. The basic idea is to parse sentences from corpora containing a verb with an unknown subcategorization frame several times, each time throwing in a different hypothetical subcategorization frame. The only sentences of interest are those for which just one hypothetical frame leads to a successful parse: here, the hypothesis is likely to be correct. Ungrammatical marks are employed to further restrict candidate sentences that go into lexicon acquisition, in order to arrive at a reasonable precision. For example, contexts like (32) which do not provide a clear distinction between a transitive and an inherently reflexive verb are excluded.

(32) Ich habe mich korrigiert.
I have myself corrected

During the acquisition experiment, the constraints that are designed to rule out such unclear examples are attached to the root category of the grammar and are marked as the only "ungrammatical" constraints (in the specific technical sense). This has the effect that sentences that cannot be parsed other than by applying such a marked constraint are returned by the system with an asterisk along with the number of solutions. This provides the designer of the experiment with important information: the sentence did not fail because it was not covered by the original grammar; instead, it was merely suppressed by the technically motivated extra constraints.

3.5 Generation

Another use of OT which is applicable to more technical contexts is to mark rules which are used only in parsing and not in generation. UNGRAMMATICAL marks are a clear instance of this since even though it may be desirable to have the grammar parse ungrammatical sentences to improve robustness, it is unlikely that one would wish to have such ungrammatical structures produced. Another use is for punctuation control: in general, it is good to be able to parse punctuation in a large number of positions but to generate it in a much more restricted domain. For example, although commas appear with reckless abandon in many texts, they can be generated in a more controlled fashion. Finally, certain constructions may be technically grammatical but not be ones that one wishes to generate. For example, it is possible to place a *when* clause after the subject and before the verb in English, as in (33a). However, the grammar could be restricted to only generate *when* clauses in sentence-initial and sentence-final positions, as in (33b).

(33) a. The rear burner, when left on for too long, will tend to overheat.

b. S $\longrightarrow$ NP CONJPsub VP

$(\uparrow\text{SUBJ})=\downarrow$ $(\uparrow\text{ADJUNCT})=\downarrow$ $\uparrow=\downarrow$

ParseOnly $\in o*$

3.6 Robustness

One of the most useful aspects of the OT projection is to increase the robustness of the grammar. There are a number of ways in which this can be done. Parsing actual text or speech unfortunately involves input which is not always strictly speaking grammatical. By marking rules which parse such input with an UNGRAMMATICAL constraint mark, it is possible for the grammar to assign the best possible analysis under the circumstances while still indicating that the sentence is in fact not well formed. An example of this type comes from gender and number mismatches in French. Normally, adjectives agree with the nouns that they modify in gender and number. However, instances like those in (34) are often found.

(34) Les moteurs sont grand.
the motors.MASC-PL are big.MASC-SG
'The motors are big.'

OT marks in the adjectival agreement rules can allow such sentences to be parsed as ungrammatical. Again, such ungrammatical constructions will be ruled out in generation mode.

Not all issues affecting robustness involve ungrammatical constructions. Sometimes, it is virtually impossible to write a grammar which encompasses all the possibilities of a construction. An example of this is the use of labels whose source can be practically any part of speech and even non-consituent combinations, as in (35).

(35) a. the save as button

b. the print button

c. the reload button

d. the fast forward button

The "label" construction in (35) is basically of the form *the X button* where X can range over a large number of parts of speech. A special c-structure rule can be written to allow for the unusual labels but to disprefer them in relation to more conventional parses to avoid odd parses of simple NPs such as *the red button*.

Another way to increase the robustness of the grammar is to allow lowly ranked catch-all rules which can combine constituents which the grammar does not otherwise handle. Due to lack of space, these will not be discussed further here.

4 Discussion and Outlook

In this paper we introduced a way to integrate OT with LFG and discussed a number of ways in which OT can be used to facilitate the writing of large scale LFG grammars. These included preference marks for specific constructions, dispreference marks for rare constructions, ungrammaticality marks to increase robustness in parsing, and NOGOOD marks to allow specialized corpus rules to be turned off. Our intuition is that OT is most appropriate for distinguishing between different degrees of grammaticality, especially between grammatical and ungrammatical analyses. This is similar to the convention of using ?, ??, *, and ** to indicate different degrees of ungrammaticality of different readings.

We discussed cases of syntactic ambiguities where the usage of OT to express syntactic preferences between otherwise grammatical analyses was problematic. One reason for this was that it is almost always possible to construct a context in which the dispreferred reading is the best one for semantic or pragmatic readings. Another reason is that OT assigns a global ranking over all analyses. This means that if you try to express a preference between two different analyses, you also end up expressing preferences between these analyses and other, completely unrelated analyses. However, in some cases it is possible to state preference constraints in a way to avoid such interaction problems.

In spite of these problems, we believe that syntactic preferences can be useful as a component of a larger scheme for disambiguation which includes information from other sources, like semantic constraints or statistical weights.

References

Bresnan, J. (1996a). Optimal Syntax: Notes on Projection, Heads, and Optimality. Ms., Stanford University.

Bresnan, J. (1996b). LFG in an OT Setting: Modelling Competition and Economy. In *Proceedings of the First LFG Conference*. CSLI Proceedings ON-LINE.

Butt, M., T.H. King, M-E. Niño, F. Segond. (to appear). *A Grammar-Writer's Cookbook*. CSLI Publications.

Kuhn, J., Eckle-Kohler, J., and Rohrer, C. (1998). Lexicon acquisition with and for symbolic NLP-systems – a bootstrapping approach. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC98)*, Granada, Spain.

Kuhn, J., and Rohrer, C. (1997). Approaching ambiguity in real-life sentences – the application of an Optimality Theory-inspired constraint ranking in a large-scale LFG grammar. In *Proceedings of DGfS-CL*, Heidelberg.

Prince, A., and P. Smolensky. (1993). Optimality Theory: Constraint Interaction in Generative Grammar. Technical Report 2, Rutgers University Center for Cognitive Science.

Anette Frank
anette.frank@xrce.xerox.com
XRCE
Grenoble Laboratory
6, chemin de Maupertius
38240 Meylan FRANCE

Jonas Kuhn
jonas@ims.uni-stuttgart.de
IMS
Universität Stuttgart
Azenbergstr. 12
D-70174 Stuttgart GERMANY

Tracy Holloway King
thking@parc.xerox.com
NLTT/ISTL
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA

John Maxwell
maxwell@parc.xerox.com
NLTT/ISTL
3333 Coyote Hill Rd.
Palo Alto, CA 94304 USA