# REVERSING F-STRUCTURE REWRITING FOR GENERATION FROM MEANING REPRESENTATIONS

Sina Zarrieß and Jonas Kuhn
Universität Stuttgart

Proceedings of the LFG10 Conference

Miriam Butt and Tracy Holloway King (Editors)

2010

CSLI Publications

http://csli-publications.stanford.edu/

**Abstract**

We describe the design of an LFG-based generation system that provides a framework for empirical studies on the choice among grammatical paraphrases (i.e. syntactic alternations), as an effect of interacting soft constraints. To be able to study the relevant variation, we extend the XLE generation architecture so it no longer departs from standard f-structures, but from a more abstract level of (meaning) representation. This representation is constructed by means of XFR term-rewrite rules. We discuss the design of the meaning representation in light of the surface realisation task. In particular, we address the problem of obtaining a transfer grammar that reverses meaning construction, taking into account the generation performance.

# 1 Introduction

In this paper, we describe the design of an LFG-based generation system that provides a framework for studying soft constraints on grammatical paraphrases, i.e. syntactic alternations. These alternations have recently attracted interest in theoretical linguistic research, motivating models of grammar that assume statistical preferences to be guiding the use of certain linguistic constructions. By way of illustration, we cite an example from Bresnan and Ford (2010):

Given the following linguistic context in a dialogue:

(1) And I said, I want a backpack.
I told him, if you want to give me a present for Christmas ...

What is the most likely continuation of the sentence?

(2) a. ... give me a backpack.
b. ... give a backpack to me.

The alternatives in (2) illustrate the English dative alternation. Bresnan et al. (2007) show that speakers prefer one over the other construction depending on the discourse context and the discourse accessibility of the verb's arguments. For (2), the speaker in the dataset chose (2-a). This can be explained by the fact that the speakers statistically prefer first-person, pronominal, discourse-given recipients (*me*) to precede nominal, discourse-new themes (*backpack*).

Interestingly, the insight that discourse properties of referents are an informative factor in modelling linguistic preferences among grammatical variants is corroborated by computational research using generation with implemented broad-coverage grammars – where the relevant distinctions are subject to complex interactions of multiple factors and information sources. Cahill and Riester (2009) use the generator integrated in the XLE system to generate syntactic alternations (mainly word order variations) from given corpus sentences. They address the task of ranking these alternations, i.e. finding the appropriate realisation in context, by training a log-linear statistical model to replicate the actual realization choices for corpus data from a treebank. Their experiments show that a model approximating discourse properties of the referents in a sentence improves the results of the realisation ranking model.

There are a number of lingustically interesting alternations that the work by Cahill and Riester (2009) could not study as participating in the ranking process, e.g. argument or voice alternations. F-structures are usually underspecified at the level of word order, but not at the more abstract semantic level encoding the realisation of predicate arguments. For instance, an LFG grammar would usually assign different f-structure representations to the active and passive realisation of a sentence. To be able to include these alternations, we need to extend the current XLE generation architecture so it departs from a level of representation abstracting away from syntactic alternations.

XLE supports generation from partially underspecified feature structure representations.[1] So, in principle, one could design a brand new feature representation for the intended level of abstraction. However, a level of representation normalising the relevant alternations has already been designed and related to f-structures from the ParGram LFG grammars, in the context of textual entailment and question answering tasks: Crouch and King (2006). Since our experiments are aimed at capturing interaction effects in real corpus data, it is important to achieve broad coverage of syntactic, morphological and lexical phenomena relatively fast. So, the most natural way to go is to adapt the existing representation and mapping mechanism for our purposes.

Crouch and King (2006) use the term-rewrite transfer system included in the XLE system (the "XFR system"), for mapping f-structures to flat semantic representations. Originally designed for machine translation, the system has proven highly useful from a practical point of view, since it supports rapid data-oriented engineering for various kinds of format conversion. The resulting transfer rule sets are generally very robust, since it is easy to include catch-all rules (and override them for specific data instances). It is also relatively straightforward to port an XFR transfer grammar from one ParGram grammar to another, taking advantage of the carefully controlled parallel f-structure geometry across languages.

The XFR system is unidirectional, so it cannot be reversed directly. This means that for our project of building semantics-based generation taking advantage of existing work on meaning construction, we have to address two questions: (1) what should be the design for our meaning representation (which parts of the entailment-oriented shallow semantics do we want to take over, etc.), and (2) how can the reverse mapping from the meaning representation to (a packed representation of all possible) f-structures be realized.

We introduce the task of surface realisation ranking in more detail and discuss the motivations of this work in Section 2. In Section 3, we provide a brief overview

[1]There are limits posed by theoretical results showing that the generation from underspecified features structures is undecidable in the general case (Wedekind, 1999) – but the XLE generator takes advantage of the constructive approach of Kaplan and Wedekind (2000).

of the extended generation architecture proposed in this paper. Section 4 describes the design of the meaning representation and discusses several adaptations for the surface realisation task. Finally, in Section 5, we treat the problem of obtaining a transfer grammar that reverses the meaning construction and point out its relation to generation performance.

# 2 Surface Realisation Ranking in the LFG Architecture

## 2.1 Ranking in the LFG Architecture

LFG grammars implemented in the XLE framework are generally reversible so that they can be used in parsing and generation. In both scenarios, one has to deal with disambiguation, i.e. ranking problems. Formally, the disambiguation problem amounts to the selection of the (or a) contextually appropriate analysis/realisation from a set of candidates that is characterised by underspecification in the shape of a "packed" LFG representation. In parsing, all candidate analyses share a common surface string; in generation, the candidate realisations share (a partial specification of) an underlying input representation, typically a partial f(unctional) structure. The two dual choice problems are illustrated on the left-hand side of Figure 1.

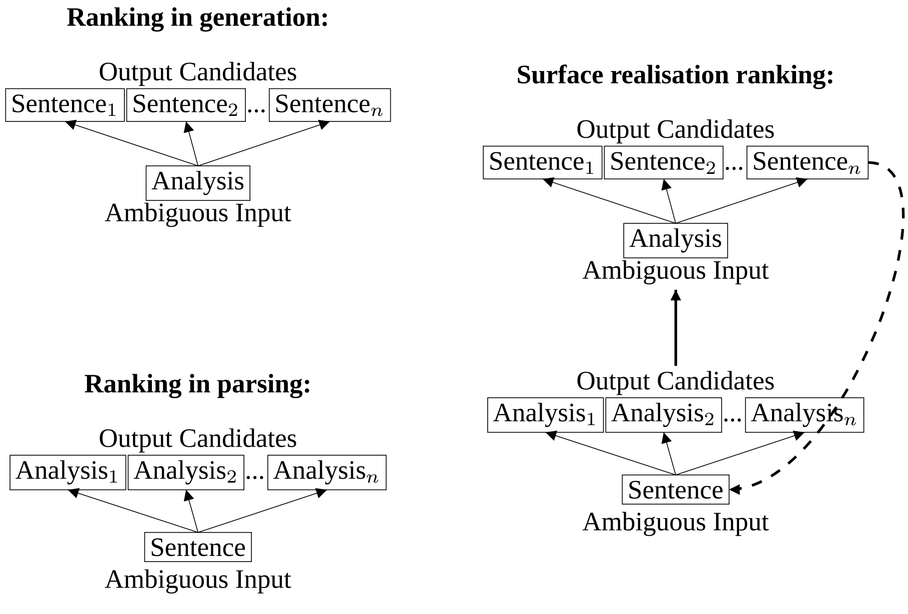


Figure 1: Ranking in a reversible grammar architecture

In both scenarios, log-linear statistical models for ranking the candidates have proven successful for modelling the preferred choice based on corpus data modelling the linguistic experience of a speaker.[2] Formally, the set-up is very similar

[2] In the log-linear ranking approach, each candidate structure is represented as a vector of its

to the Optimality Theoretic (OT) LFG architecture (Bresnan, 1996, 2000), which can be based on the same reversible framework of candidate generation (Kuhn, 2000, 2001, 2003).[3]

From a generation perspective, an LFG f-structure can be considered an abstract syntactic representation that is underspecified with respect to word order and certain aspects of morphological/lexical word choice (Cahill et al., 2007). Therefore, in mapping f-structures onto surface sentence strings, one usually obtains various possible, truth-conditionally equivalent realisations.

The quality of a mechanism choosing a particular surface realisation can be evaluated straightforwardly in a corpus-based setting. The typical design for testing a surface realisation component against realistic corpus data is displayed on the right-hand side of Figure 1 (going from the bottom to the top): First, a corpus sentence is parsed and mapped to a linguistic, underspecified analysis. Second, a generator maps this analysis to all possible surface realisations which have to be ranked by a realisation ranking model. Finally, the output of the ranking is compared against the original corpus sentence. There are multiple ways and measures to assess this comparison, such as automatic measures from Machine Translation evaluation or human judgements (Cahill, 2009).

## 2.2 Ranking for Free Word Order Languages

The ranking problems described in the previous section are especially challenging in languages with free word order. Consider the following example from German:

| | | | | | |
|---|---|---|---|---|---|
| (3) | Maria | schenkt | Thomas | ein | Buch. |
| | Maria.NOM | gives | Thomas.DAT | a | book.ACC. |

Sentence (3) illustrates a verb with three case-ambigous arguments, such that in parsing the sentence receives four possible analyses (*Mary* can be the subject, and the direct and indirect object, *Thomas* can also have all three functions, *book* can be the subject and the direct object). When we generate from an arbitrary f-structure for Sentence (3), we obtain the set of surface realisations in (4) which amounts to the set of all permutations of the three arguments.

(4) a. Maria schenkt ein Buch Thomas.
b. Maria schenkt Thomas ein Buch.
c. Ein Buch schenkt Thomas Maria.
d. Ein Buch schenkt Maria Thomas.

contextually relevant properties. The property weights (corresponding to the relative ranks of the constraints in an OT setting) can be discriminatively trained on corpus data using numerical optimization algorithms, which ensure that the weights for the various properties are set in such a way that the observed analyses/realisations are ranked the highest (Riezler et al., 2002; Cahill et al., 2007).

[3]The close relationship between an OT constraint ranking approach and log-linear models (which is just a different name for Maximum Entropy models) is discussed by Goldwater and Johnson (2003) and Jäger (2004).

e. Thomas schenkt ein Buch Maria.
f. Thomas schenkt Maria ein Buch.

If we were able to generate from an f-structure underspecified for voice, we would additionally obtain the surface realisations in (5) illustrating all possible permutations in passive voice (where in German only the theme argument can be turned into the passive subject).

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| (5) | a. | Maria | wird | von | Thomas | ein | Buch | geschenkt. |
| | | Maria.DAT | is | by | Thomas | a | book.NOM | given. |

b. Maria wird ein Buch von Thomas geschenkt.
c. Ein Buch wird Maria von Thomas geschenkt.
d. Ein Buch wird von Thomas Maria geschenkt.
e. Von Thomas wird Maria ein Buch geschenkt.
f. Von Thomas wird ein Buch Maria geschenkt.

To our knowledge, the impact of syntactic alternations like voice on realisation ranking in free word order languages has so far not been investigated in computational frameworks working with reversible grammars. Velldal (2008) reports on HPSG-based generation experiments for English where he constrasts generation from meaning representations that are underspecified and specified for voice and topicalisation. As one would expect, the underspecified representations trigger much more (about twice as many) surface realisation candidates and the ranking task becomes much harder.

While it is difficult to compare surface realisation experiments based on different grammars and languages, one would, at least theoretically, expect that the status or function of syntactic alternations differs between languages like English and German, since German has more options available for achieving a particular ordering and hence, conveying subtle information structural differences. In English, the use of syntactic alternations (e.g. the dative alternation) is often attributed to statistical word order patterns. Bresnan et al. (2007) base their explanation of the dative alternation on the finding that “animate, pronominal, short, discourse-acessible arguments tend to precede inanimate, nonpronominal, long arguments.” In German, the situation is less clear, since these precedence patterns are not constrained by the word order restrictions.

### 2.3 Surface Realisation and the Problem of Input Representation

Before moving on to the design of the extended generation architecture, we briefly point out an additional, independent advantage of using a more abstract shallow meaning representation instead of a standard LFG f-structure.

Grammar-based generators are a good basis for focussed studies on surface realisation (or “tactical” generation), since these systems (usually) produce grammatical output, and are actually able to produce all grammatical realisations of a given abstract input. However, an obvious limitation of grammar-based generators is that they require a very specific input representation which corresponds to the internal specification of the grammar. Depending on the system context of the surface realiser, this input representation is often hard (or almost impossible) to predict in external applications (see Section 4). As a consequence, grammar-based generators are rarely used in real-life generation applications. A further disadvantage of grammar-specific input for generation is the fact that the results obtained by different generators based on different grammars or input representations are difficult to compare (Belz et al., 2010).[4]

We extend the XLE generation set-up to take a more shallow representation as input, using an added conversion step at the beginning. This can be seen as a first step towards making the grammar-based XLE generator applicable in traditional NLG domains, like e.g. text summarisation, where the input representation can be expected to be more abstract or underspecified than fully-fledged LFG f-structures. The initial conversion step can be re-engineered fairly easily to adapt it to the relevant system context.

## 3 System Overview

The work presented in this paper investigates the feasibility of interfacing the XLE generator with a preprocessing step, which produces a packed underspecified f-structure representation of the f-structures compatible with a shallow meaning representation, abstracting away from morpho-syntactic alternations. As pointed out in Section 1, practical engineering considerations lead us to assume that this shallow input representation is most conveniently built by means of transfer rules, reusing a good deal of the work on meaning representations in Crouch and King (2006), a.o.

The generation architecture we propose is illustrated in Figure 2. First, an input corpus sentence is parsed and mapped to a flat semantic representation. Note that the subject of the passive f-structure is mapped to a "semantic object" in the meaning representation. In the reverse mapping from meaning representation to f-structures, the generator produces an f-structure chart that, besides the original f-structure, realises its meaning-equivalent syntactic paraphrases, e.g. voice alternations. This f-structure chart is then mapped to all its corresponding surface sentences by means of the standard XLE generator. Finally, a ranking model selects the most appropriate surface realisation.

Thus, our surface realisation testing architecture is very similar to Cahill et al. (2007). We just extend their generation pipeline by intermediate steps of further

[4]One reason for the lack of comparable tools for surface realisation is the lack of standardised resources annotated with semantic representations. Bohnet et al. (2010) present statistical generation experiments on the CoNLL'09 data which integrates semantic annotations from PropBank. However, they face the problem that this semantic annotation is far from complete, i.e. the relations between certain words are missing (e.g. adjectival modifiers). As a solution, Bohnet et al. (2010) add the missing semantic relations based on some handcrafted rules and the underlying dependency tree which results in semantic representations very similar to syntactic representations.

‘Tom was discovered by Chomsky.’

$\downarrow$ *XLE parsing*

*Original F-structure:*

$$\left[\begin{array}{ll}
\text{PRED} & '\text{discover} < (\uparrow \text{SUBJ})(\uparrow \text{OBL-AG}) >' \\
\text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & '\text{Tom}' \\ \text{NTYPE} & \text{proper-name} \\ \text{PERS} & 3 \end{array}\right] \\
\text{OBL-AG} & \left[\begin{array}{ll} \text{PRED} & '\text{Chomsky}' \\ \ldots & \end{array}\right] \\
\text{TOPIC} & [\text{'Tom'}] \\
\text{VTYPE} & \text{main} \\
\text{PASSIVE} & +
\end{array}\right]$$

$\downarrow$ *transfer meaning construction*

| HEAD (discover) |
|---|
| PAST (discover) |
| ROLE (sem-subj,discover,Chomsky) |
| ROLE (sem-obj,discover,Tom) |

$\downarrow$ *reverse transfer mapping*

*Output F-structure Chart:*

$$\left[\begin{array}{l}
\left[\begin{array}{ll}
\text{PRED} & '\text{discover} < (\uparrow \text{SUBJ})(\uparrow \text{OBJ}) >' \\
\text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & '\text{Chomsky}' \end{array}\right] \\
\text{OBJ} & \left[\begin{array}{ll} \text{PRED} & '\text{Tom}' \end{array}\right]
\end{array}\right] \\
\\
\left[\begin{array}{ll}
\text{PRED} & '\text{discover} < (\uparrow \text{SUBJ})(\uparrow \text{OBL-AG}) >' \\
\text{SUBJ} & \left[\begin{array}{ll} \text{PRED} & '\text{Tom}' \end{array}\right] \\
\text{OBL-AG} & \left[\begin{array}{ll} \text{PRED} & '\text{Chomsky}' \end{array}\right]
\end{array}\right]
\end{array}\right]$$

$\downarrow$ *XLE generation*

“Tom was discovered by Chomsky.”
“Chomsky discovered Tom.”

$\downarrow$ *surface realisation ranking*

“Tom was discovered by Chomsky.”

Figure 2: Generation via meaning representations

analysis, followed by generation of a broader f-structure (chart) representation, such that we do not directly regenerate from an f-structure obtained for a corpus sentence. This work focuses on the added intermediate steps in the testing pipeline, i.e. the bidirectional mapping between f-structures and a more abstract meaning representation. We leave examination of the final realisation ranking in the new setting for future work.

By using the XLE grammar-based generator in our architecture, we make sure that the final output of the generation system (if there is one) is a grammatical sentence. However, it is important to note that in the additional generation step from semantics to f-structure (charts), the wellformedness of the transfer output is not guaranteed or checked since the transfer rules can produce arbitrary sets of f-structure terms as output. We use the grammar-based generator as a filter (similar to Crouch et al. (2004)) that only maps those f-structures to surface sentences that correspond to the definition of the underlying grammar.

# 4 A Meaning Representation for Surface Realisation

The standard meaning construction approach for the ParGram LFG grammars implemented in the XLE framework is the transfer semantics system developed by Crouch and King (2006). It has been ported to German by Zarrieß (2009). The system exploits the XLE transfer module to map LFG f-structures to shallow meaning representations on the basis of an ordered list of term-rewrite rules. In this section, we will discuss the design of the representation and its usefulness for generation. In the next section, we will discuss the technical aspects of reversing the rules for generation.

## 4.1 Normalising Paraphrases

The main purpose that a meaning representation for surface realisation serves is to normalise the analyses of truth-conditionally equivalent syntactic structures. As the result of this normalisation, syntactic alternations get assigned an identical meaning representation. In the generation step, the surface realiser will then map the meaning representation to all its possible syntactic alternations.

The semantic representation we want to generate from was originally designed for a textual entailment application (Crouch and King, 2006). To capture the entailment relation between, e.g., active and passive realisations of a verb, the representation assigns a uniform analysis to these alternations. As an example, the sentences in (6-a) and (6-b) would both be assigned the meaning representation in (6-c). The subject of the active and the oblique agent of the passive verb are both normalised to a “semantic subject”. Such a normalised meaning representation is exactly what we need in generation.

(6) a. Peter saw Mary.
b. Mary was seen by Peter.
c.

| HEAD `(see)` |
|---|
| PAST `(see)`<br>ROLE `(sem-subj,see,Peter)`<br>ROLE `(sem-obj,see,Mary)` |

The meaning construction mechanism from Crouch and King (2006) implements a number of further normalisation operations for other types of paraphrases or alternations that are both interesting for entailment and surface realisation. The implemented normalisations include the following:

(7) a. Attributive vs. predicative modifiers
- (i) Peter reads a good book.
- (ii) Peter reads a book that is good.

b. Clefts
- (i) It is a book that Peter reads.
- (ii) Peter reads a book.

c. Genitives
- (i) the building’s shadow
- (ii) the shadow of the building

d. Nominalisations vs. verbal realisations
- (i) Peter regrets the destruction of the city.
- (ii) Peter regrets that the city was destroyed.

## 4.2 Implicit Syntactic Information

An important aspect of the paraphrase normalisation is that many syntax-internal features are removed from the meaning representation. In practice, the f-structures that correspond to a certain pair of meaning-equivalent sentences, e.g. active and passive alternations, do not only differ in their argument frame and passive feature. The f-structures usually also specify a lot of other, e.g. morphological, features of the involved noun phrases and the verb that differ between the alternation constructions. An example alternation pair and its corresponding f-structure pair is given in Table 1.

| | |
|---|---|
| Tom sieht Marie.<br>Tom sees Mary. | $\begin{bmatrix} \text{PRED} & '\text{sehen} < (\uparrow ...)(\uparrow ...) >' \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & '\text{Tom}' \\ \text{CASE} & \text{nom} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & '\text{Marie}' \\ \text{CASE} & \text{acc} \end{bmatrix} \\ \text{CHECK} & \begin{bmatrix} \text{AUX-SELECT} & '\text{haben}' \end{bmatrix} \\ \text{TOPIC} & \begin{bmatrix} '\text{Tom}' \end{bmatrix} \\ \text{PASS} & - \end{bmatrix}$ |
| Marie wird von Tom gesehen.<br>Mary is by Tom seen. | $\begin{bmatrix} \text{PRED} & '\text{sehen} < (\uparrow ...)(\uparrow ...) >' \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & '\text{Marie}' \\ \text{CASE} & \text{nom} \end{bmatrix} \\ \text{OBL-AG} & \begin{bmatrix} \text{PRED} & '\text{Tom}' \\ \text{CASE} & \text{dat} \end{bmatrix} \\ \text{CHECK} & \begin{bmatrix} \text{AUX-SELECT} & '\text{sein}' \\ \text{PARTICIPLE} & '\text{perfect}' \end{bmatrix} \\ \text{TOPIC} & \begin{bmatrix} '\text{Marie}' \end{bmatrix} \\ \text{PASS} & + \end{bmatrix}$ |

Table 1: F-structure pair for passive-active alternation

For generation, it is important to remove these syntax-internal features because they would implicitly disambiguate an abstract semantic representation of an alternation. For instance, if the meaning representation would not underspecify the case of a noun phrase, the surface realiser would have implicit syntactic information about the original sentence realisation.

In the LFG ParGram grammars, many of these syntax-internal features are conventionally subsumed under the technically motivated CHECK-feature. So these can easily be detected and removed when constructing the semantic representation. In the mapping from semantics to f-structure, these features do not need to be reconstructed since the XLE generator can deal with underspecified f-structures (see Section 5.1).

However, in certain problematic cases, the f-structures for an alternation pair contain implicit features that are not syntax-internal. As an example, consider the sentence pair in Table 2. The analyses are produced by a German LFG grammar whose lexicon does not have an entry for the proper noun *Karthago*. XLE provides a "guessing" mechanism for unknown words. In this case, the German grammar has been set up to assume that unknown capitalized word forms are proper names, leaving the gender and number feature unspecified (since there are proper names for all genders and in singular and plural – like *Beatles*).

As a consequence, the f-structure for *Karthago* in the passive sentence does not have a NUM feature since the number of the noun cannot be inferred from the syntax. By contrast, the f-structure for *Karthago* in the active sentence does have a NUM feature which comes from the inflectional morphology of the verb. So the two sentences have different meaning representations (if the meaning construction takes number into account).

Such types of implicit information in the f-structure are not easy to deal with in generation. First, it is difficult in practice to foresee such problems and debug them when they occur. Second, the XLE generator is very sensitive to slight changes in the f-structure input. If the surface realiser were to add a NUM feature to the f-structure in the passive sentence in Table 2 (which may seem to be a reasonable move), the generator would fail (because the structure that the grammar assigns to the sentence is no longer subsumed by the input representation). On the other hand, one would drastically change the output of the surface realisation if the NUM feature was generally underspecified (in this case, the generator would produce the singular and plural realisation for each noun phrase in a given input f-structure).[5]

While the above type of grammar-internal, implicit information may suggest we are dealing with more of a technical than a principled problem, similar cases of indirect disambiguation of a meaning representation *do* occur in situations that are

[5]This problem with syntax-internal, atomic features has also been noted in other applications, e.g. Machine Translation. Graham (2010) reports drastically varying performance of their MT system depending on the quality of atomic feature translation. She also reports that grammar coverage of the generator varies between 12% and 41% depending only on the translation quality of the atomic features. This corroborates the aforementioned claim that grammar-based generators can be hard to use in external applications.

| | |
|---|---|
| Rom wurde von Karthago erobert.<br>Rome was by Carthage conquered. | $\begin{bmatrix} \text{PRED} & '\text{erobern} < (\uparrow ...)(\uparrow ...) >' \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & '\text{Rom}' \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix} \\ \text{OBL}_{AG} & \begin{bmatrix} \text{PRED} & '\text{Karthago}' \\ \text{PERS} & 3 \end{bmatrix} \\ \text{PASS} & + \end{bmatrix}$ |
| Karthago eroberte Rom.<br>Carthage conquered Rome. | $\begin{bmatrix} \text{PRED} & '\text{erobern} < (\uparrow ...)(\uparrow ...) >' \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & '\text{Karthago}' \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & '\text{Rom}' \\ \text{PERS} & 3 \\ \text{NUM} & \text{sg} \end{bmatrix} \\ \text{PASS} & - \end{bmatrix}$ |

Table 2: F-structure pair for passive-active alternation: the features for *Karthago* are asymmetric

fully motivated linguistically. These structures need to be addressed in the meaning construction. For (8-a), the normalised meaning representation (8-b) contains implicit information that its original sentence must have been realised in active voice. This is because the subject of the sentence is the generic pronoun *man* which cannot be used as an oblique agent in a prepositional phrase, i.e., (8-c) is ungrammatical. Thus, if the realiser derives an f-structure where the generic pronoun is realised as the oblique agent, the grammar-based generator rules will not produce a surface sentence for this input.

(8) a. Man hat Maria im Park gesehen.
One has Mary in the park seen.

b.

| HEAD `(see)` |
|---|
| PAST `(see)` |
| ROLE `(sem-subj,sehen,man)` |
| ROLE `(sem-obj,sehen,Maria)` |

c. *Maria wurde von man im Park gesehen.
Mary was by one in the park seen.

d. Maria wurde von jemandem im Park gesehen.
Mary was by somebody in the park seen.

In order to be able to generate a passive paraphrase from Sentence (8-a), the meaning representation would have to abstract away from the lexical realisation of the pronoun such that the generator could realise the subject as a different pronoun, e.g. *jemand* (*somebody*), as in (8-d). As a consequence, the surface realisation step would be extended from word order and structural choice to lexical choice, which is usually considered as a separate step of generation (Bateman and Zock, 2003).

A similar and very frequent type of implicit syntactic information occurs in coordinated sentences. For instance, in sentence (9), the noun phrase *Tom* is the subject of two verb phrases. At the moment, the meaning representation keeps the information about the lexical identity of the two subjects in a lexical index (marked as integers in (9-b)). If the generator "knows" that the two subjects have to be realised by the same noun phrase, it cannot produce a passive paraphrase for one of the verb phrases due to syntactic constraints. However, if we interpret the representation as even more abstract and allow the realiser to generate a pronoun for *Tom* in one of the verb phrases (such as in (9-c)), we introduce a completely new type of generation problem (i.e. the generation of referring expressions) into our system.

(9) a. Tom sieht Marie und schenkt ihr einen Apfel.
Tom sees Marie and gives her an apple.
b.

| HEAD (sehen) |
|---|
| ROLE (sem-subj,sehen,Tom:1) |
| ROLE (sem-obj,sehen,Marie:2) |
| ROLE (sem-subj,schenken,Tom:1) |
| ROLE (sem-obj,schenken,Apfel:3) |
| ROLE (recipient,schenken,sie:4) |

c. Marie wurde von Tom gesehen und bekam von ihm einen Apfel geschenkt.
Marie was by Tom seen and got by him an apple given.

Finally, the type of implicit syntactic information that needs to be added or removed in paraphrase normalisation is also dependent on the complexity of the underlying alternation. For instance, the meaning representation normalises relative clauses and deverbal attributive adjuncts, such as (10-a-b). However, the non-finite verb in (10-a) does not carry any tense information whereas the finite verb in (10-b-c) does. Thus, in order to generate a relative clause paraphrase for (10-a), the meaning construction needs to include rules that infer the tense of *laughing*.

(10) a. Peter saw a laughing girl.
b. Peter saw a girl who was laughing.
c.

| HEAD (see) |
|---|
| PAST (see) |
| PAST (laugh) |
| ROLE (sem-subj,see,Peter) |
| ROLE (sem-obj,see,girl) |
| ROLE (sem-subj,laugh,girl) |

All these examples show that the boundaries between lexicalisation, grammaticalisation and surface realisation in generation get blurred rather quickly. Thus, the design decisions made at the level of meaning representation will greatly influence the difficulty and the outcome of the final surface realisation task. Moreover, we have seen that the meaning representations and the f-structures of a given alternation pair have to be carefully examined in a variety of syntactic contexts in order to produce well-formed input for the grammar-based generator.

## 5 Reversing Meaning Construction

This section addresses the issue of mapping meaning representations as discussed in Section 4 to f-structure chart representations from which the standard XLE generator is able to generate – given the fact that the XFR system is not directly reversible. We already mentioned the engineering advantage of re-using existing resources as much as possible – in particular in view of the multilingual setting of ParGram, which will make it relatively easy to port solutions to other languages. Hence, our approach is to develop XFR rules for the backward mapping from meaning representations to f-structures that draw upon the forward mapping rules as much as possible.[6] In Section 5.1, we show that if the meaning construction is restricted to a specific type of normalisation rules and if the generation of syntax-internal features is left to the grammar-based generator, the reverse transfer grammar can be easily derived.

A second important issue raised by our surface realisation architecture is the computational complexity and runtime performance of generation. The f-structure output produced by a reverse meaning construction is formally more complex than the f-structures that have been used in surface realisation experiments so far: whereas Cahill et al. (2007) generate from single f-structures that represent one possible analysis of a sentence, we will generate from f-structure charts which represent all the possible realisations of a syntactic alternation. Moreover, the f-structures used by Cahill et al. (2007) are almost completely specified, i.e., they contain all the syntax-internal features needed by the grammar. In our case, as already mentioned in Section 4, the f-structures will necessarily by underspecified to a certain degree since not all syntax-internal features can and should be reconstructed from the meaning representation. These properties of the f-structure input will have a noticeable effect on generation performance, which we will discuss in Section 5.2.

### 5.1 Transfer Rules and Bidirectionality

The XFR term rewrite system has been used in a variety of system contexts: f-structure based machine translation (Riezler and Maxwell, 2006), sentence condensation (Crouch et al., 2004), and textual entailment oriented shallow meaning construction (Crouch and King, 2006). See Crouch et al. (2004) for a detailed illustration of the XFR system.

According to Emele et al. (1996), term rewrite rules can be defined as follows:

(11) a. $\langle$*LHS Set*$\rangle\#\langle$*LHS Conds*$\rangle \leftrightarrow \langle$*RHS Set*$\rangle\#\langle$*RHS Conds*$\rangle$
b. $\langle LHS\ Set\rangle\#\langle$*LHS Conds*$\rangle \rightarrow \langle$*RHS Set*$\rangle$

[6] As an alternative option, one could consider a system that automatically learns the mapping between these structures, in the style of Bohnet et al. (2010). However, we feel that such a purely statistical approach ignores much of the implicit knowledge given in the forward meaning construction grammar and that it risks producing output incompatible with the XLE generator.

c. $\langle LHS\ Set\rangle \leftarrow \langle$*RHS Set*$\rangle \# \langle$*RHS Conds*$\rangle$

The most general definition in (11-a) specifies a transfer rule as a bidirectional rewrite relation between a set of left hand side terms and a set of right hand side terms. The rewrite can be conditioned on a set of terms on both sides of the rule. The corresponding unidirectional rule definitions are given in (11-b-c). In a unidirectional transfer rule, only one rule side can have rewrite conditions.

The XFR system represents an f-structure internally as a set of two-place terms.[7] By this means, one can formulate rewrite rules on f-structures that perform arbitrary lexical and structural transformations. An example rewrite rule is given in (12). The sample rule applies to f-structures that have a PASSIVE and VTYPE feature as well as an oblique agent, mapping the oblique agent to a "logical subject" (i.e., using the f-structure of active clauses as the prototypical representation).

(12) $+\mathit{VTYPE}(\%V, \%\%), +\mathit{PASSIVE}(\%V, +), \mathit{OBL\text{-}AG}(\%V, \%\mathit{LogicalSUBJ})$
$==> \mathit{SUBJ}(\%V, \%\mathit{LogicalSUBJ}).$

As a unidirectional system, the XFR syntax allows conditions only on the left hand side of rules. Other transfer systems, such as Emele et al. (1996) from the Verbmobil project, implement a bidirectional syntax for rewrite rules. However, Emele et al. (1996) also mention that the implementation of a bidirectional transfer grammar is difficult in the case of large sets of rules. They report that unidirectional rules are more effective in practice since the grammar writer does not have to keep track of the bidirectional rule conditions.

In the case of meaning construction, it would presumably be even more difficult to specify bidirectional rewrite rules than for machine translation. One reason is that the meaning construction deletes a lot of syntax-internal features from the f-structure, e.g., CASE, PERS, or TOPIC (see the discussion on syntax-internal features in Section 4). An example for such a deletion rule is given in (13). The rule simply deletes every CASE feature from its input.

(13) $\mathit{CASE}(\%\%, \%\%) ==> 0.$

A bidirectional version of the deletion rule in (13) would have to be much more elaborate since it would need to specify exactly the contexts in which a CASE feature appears in an f-structure (essentially duplicating constraints from the grammar and the lexicon). Similarly, when we want to reverse unidirectional meaning construction rules at a fully general level, we cannot expect to find an automatic procedure that uses only the information in the forward rules.

[7] The term's name represents the f-structure attribute; the first argument is the f-structure under which the attribute is embedded (where f-structures are referenced by variables var(0), var(1), ..., which have a fixed reference for the full analysis); the second argument is the attribute value, either an atomic value (e.g., `CASE(var(1),acc))`, or an embedded f-structure node `OBJ(var(0),var(1))`. The rule syntax for terms to be rewritten vs. conditions is as follows: A prefixed $+$ on left hand rule side turns a term into a (positive) condition, which is not consumed during rule application. Identifiers starting with a % are variables.

Instead of deriving the formally exact reverse counterpart of the meaning construction transfer, we opt for an approximate transfer reversal. We do not need to generate full-fledged f-structures from the meaning representations because the XLE generator can handle underspecified input (Crouch et al., 2004) and will use the appropriate constraints from the grammar and lexicon to navigate the space of possibilities. By allowing the generator to add CASE features with arbitrary values, it can essentially follow the exact grammatical and lexical restrictions on this feature. We thus avoid a redundant (and presumably error-prone) duplication of this knowledge in the backward rewrite rules.

Leaving the generation of syntax-internal features to the generator, the general problem of reversing normalisation transfer rules is substantially simplified. As an example, consider the three rules (14). This is a typical rule set for normalisation: several sets of left hand terms, which correspond to meaning-equivalent syntactic structures, are mapped to an identical set of right hand terms. The normalisation rules in (14-a-b) are conditionned on the syntax-internal PASSIVE feature (in (14-a) it has to have the value $-$, in (14-b) the value $+$). After normalisation, the syntax-internal feature is deleted in (14-c).

(14) a. $+\mathit{PASSIVE}(\%V, -), \mathit{SUBJ}(\%V, \%\mathit{SUBJ})$
$==> \mathit{AGENT}(\%V, \%\mathit{SUBJ}).$
b. $+\mathit{PASSIVE}(\%V, +), \mathit{OBL\text{-}AG}(\%V, \%\mathit{SUBJ})$
$==> \mathit{AGENT}(\%V, \%\mathit{SUBJ}).$
c. $\mathit{PASSIVE}(\%\%, \%\%) ==> 0.$

Given that we do not need to reconstruct the syntax-internal features in the mapping from semantics to f-structure, one can straightforwardly derive a reverse version of the transfer rule sequence in (14), which is given in (15). The set of terms corresponding to the normalised partial meaning representation is optionally mapped to all its possible syntactic realisations (the $?=>$ operator stands for optional rewrite). The deletion rule in (14-c) and the rule conditions in (14-a-b) can be ignored.

(15) a. $\mathit{AGENT}(\%V, \%\mathit{SUBJ})\ ?=> \mathit{SUBJ}(\%V, \%\mathit{SUBJ})$
b. $\mathit{AGENT}(\%V, \%\mathit{SUBJ})\ ?=> \mathit{OBL\text{-}AG}(\%V, \%\mathit{SUBJ})$

Of course, in the general case, the transfer rules used for meaning construction from f-structures are not constrained to the format exemplified in (14). The grammar implemented by Crouch and King (2006) is actually far more complex and notably integrates recursive rules that rearrange the embeddings of the f-structure nodes. However, for our current work we can restrict attention to the type of simple normalisation rules, essentially a subset of the rules used by Crouch and King (2006).[8]

[8] We also implemented inspection tools for keeping track of the flow of information during term rewrite transfer, in order to isolate the relevant rules quickly.

## 5.2 F-Structure Charts in Transfer and Generation

Having discussed a way of constraining transfer rules for easy reversal, we show in this section that we need even stricter constraints on the transfer rules in order to keep the generation feasible with respect to performance.

In the reverse mapping from meaning construction to f-structure, nothing guarantees that we actually generate an f-structure that is within the coverage of a given LFG grammar. In our generation architecture (Figure 2), we rely on the fact that the XLE generator will select from the chart those f-structures that comply with the grammar specification. However, if the generator has to deal with f-structure charts that comprise a huge number of f-structures that cannot be generated from, it will often time out or fail.

By way of illustration, we contrast generation from an identical meaning representation based on two different reverse transfer grammars that generate active and passive alternations for transitive and ditransitive verbs.[9]

Our meaning representation here is simply an f-structure that abstracts from the voice of the verb, i.e. predicate arguments are mapped to semantic roles, and passive and verb morphology features are deleted from the f-structure. Depending on the formulation of the normalisation rules, the reverse generation rules may potentially look very different. In (16) and (17), we present excerpts from two transfer grammars that perform the same f-structure mappings in different ways. The transfer grammar in (16) incorporates a notion of argument frames: the semantic roles are not mapped to syntactic roles independent of each other. The naive reverse grammar in (17) on the other hand *does* employ an independent mapping rule for each semantic role.

(16) a. $\mathit{AGENT}(\%V, \%\mathit{Agent}), \mathit{THEME}(\%V, \%\mathit{Theme}),$
$\mathit{RECIPIENT}(\%V, \%\mathit{Recipient})$
?=> $\mathit{SUBJ}(\%V, \%\mathit{Agent}), \mathit{OBJ}(\%V, \%\mathit{Theme}),$
$\mathit{OBJ\text{-}TH}(\%V, \%\mathit{Recipient}).$

b. $\mathit{AGENT}(\%V, \%\mathit{Agent}), \mathit{THEME}(\%V, \%\mathit{Theme}),$
$\mathit{RECIPIENT}(\%V, \%\mathit{Recipient})$
?=> $\mathit{OBL\text{-}AG}(\%V, \%\mathit{Agent}), \mathit{SUBJ}(\%V, \%\mathit{Theme}),$
$\mathit{OBJ\text{-}TH}(\%V, \%\mathit{Recipient}).$

---

[9]Note that in German, there are two types of passive that a ditransitive verb can undergo: (1) regular passive, turning the direct object into the passive subject, and (2) *bekommen* passive, turning the indirect object into the passive subject. In the latter case, the passive is constructed with the special auxiliary *bekommen* (lit. "get"); see Example (i).

(i) a. Die Frau schenkt Maria ein Buch.
The woman.NOM gives Maria.DAT a book.ACC.

b. Ein Buch wird Maria von der Frau geschenkt.
A book.NOM is Maria.DAT by the woman given.

c. Maria bekommt ein Buch von der Frau geschenkt.
Maria.NOM gets a book.NOM by the woman given.

(17) a. *AGENT*$(\%V, \%$*Agent*$)$ ?=> *SUBJ*$(\%V, \%$*Agent*$)$.
b. *AGENT*$(\%V, \%Agent)$ ?=> *OBL-AG*$(\%V, \%AG)$
c. *THEME*$(\%V, \%$*Theme*$)$ ?=> *OBJ*$(\%V, \%$*Theme*$)$.
d. *THEME*$(\%V, \%$*Theme*$)$ ?=> *SUBJ*$(\%V, \%$*Theme*$)$.

Grammar (16) will mostly produce f-structures that are well-formed and that can be generated from, whereas grammar (17) will produce a lot of f-structures that are not compatible with LFG assumptions or specific grammatical/lexical constraints, e.g., f-structures with two subjects or without a subject. In the final surface realisation, these f-structures will not produce any surface sentence; however they substantially slow down the generation process.

For our generation experiment, we considered a set of 156 German sentences extracted from the HGC, a huge German corpus of newspaper text.[10] In Table 3, we report generation performance based on two different inputs for the surface realiser, one that was produced by means of the naive transfer rules in (17), and one that was produced by means of the lingustically informed rules in (16). The timeout parameter was set to 500 seconds. As can be seen, the generator cannot easily deal with the f-structure chart input that contains a lot of illformed structures. It times out in 30% of the cases and the average generation time is dramatically increased compared to generation from mostly well-formed input.

| | # f-structures | avg. generation time (excl. timeouts) | # timeouts |
|---|---|---|---|
| Naive Rules | 156 | 246.14 (110.68) | 53 |
| Informed Rules | 156 | 36.20 (27.04) | 3 |

Table 3: Generation performance depending on the transfer rules that produced the f-structure input

These results add an important aspect to the discussion about transfer grammar reversibility in Section 5.1. Even if we had a method that could automatically reverse any given transfer grammar, the f-structure charts produced by that reverse grammar would not necessarily be usable in generation experiments on actual corpus sentences.

Moreover, in Table 4, we compare the number of surface realisations that are produced in generation from meaning representations and generation from usual f-structures. In both cases, the total average of surface realisations is very high due to some very long sentences in our test set. If we compare the number of realisations sentence-wise, the picture is more realistic: In generation from meaning representations that abstract from the voice of a verb, the number of realisations increases by a factor of 2.8 on average. However, in 40% of the sentences, the number of surface realisations did not increase at all, i.e. no alternations could be

[10]All contain a ditransitive verb that instantiates its three arguments, such that it should generally be possible to generate several voice alternations. We did not include special rules for specific constructions like coordinations or generic pronouns (see Section 4), such that, in these contexts, the grammar will rule out the automatically generated alternations.

generated. This suggests that a more abstract meaning representation (as discussed in Section 4) would have a huge impact on the surface realisation output.

| | |
|---|---|
| Avg. number of realisations for semantic input | 25092.16 |
| Avg. number of realisations for syntactic input | 14168.57 |
| Avg. increase of realisations per sentence | 284% |
| Sentences with no increase in realisations | 64 |
| Total number of sentences | 156 |

Table 4: Number of surface realisations produced in generation from meaning representations

# 6 Conclusion

In Sections 1 and 2, we outlined the two main motivations for implementing an LFG-based surface realisation system that generates syntactic alternations from meaning representations. First, this generation architecture provides a framework for studying the interplay of multiple soft constraints on the basis of complex corpus data, taking advantage of high-quality linguistic grammars that have broad coverage at the same time. Hence, a topic of great theoretical linguistic interest can be addressed from a computational perspective. Second, this work has demonstrated the usability of the grammar-based XLE generator in a setting where the (underspecified) input representation is not directly produced by the grammar, thus taking a first step towards making the generator applicable in a wider range of natural language generation domains.

In light of the discussions and experiments presented in this paper, we can conclude that our architecture is definitely suited for carrying out targeted linguistic studies of a well-delimited set of syntactic alternations. For instance, with the help of our system, it is possible to do large-scale surface realisation experiments focussing on specific phenomena, comparing them to the smaller-scale and more controlled experiments in theoretical linguistic research, e.g. by Bresnan et al. (2007). It is also possible to empirically study the complex interaction of two or three factors known to play a role in surface realisation, e.g., word order, voice and discourse status of argument phrases.

In addition, Section 4 on the design of the meaning representation showed that by doing actual surface realisation studies, it is more likely that residual issues with a particular level of abstraction chosen as the input representation will be brought to our attention. An example is the implicit exclusion of a passive realization due to a particular lexical choice for the agent argument, or the question whether or not a tense feature is included in the abstract input representation.

Concerning the second motivation, our conclusion is more cautious. In Section 4 and 5, we have seen several difficulties with the mapping between a (more or less) grammar-external meaning representation and an f-structure input that can be dealt with by the XLE generator. The main problems are that (a) the generator is very sensitive to slight changes in the f-structure input and the underspecification mechanism does not always remedy this problem, and (b) the generator can be used to filter illformed f-structures. However, if the input contains a massive number of illformed structures, the performance decreases dramatically. In the case of well-delimited linguistic studies, both of these rather technical problems can be addressed through careful manual design of the transfer rules that map between semantics and f-structure. However, interfacing the grammar-based generator with an arbitrary semantic representation seems to require a more elaborate generation architecture.

## References


Bateman, John, and Michael Zock. 2003. Natural Language Generation. In R. Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*. Oxford University Press.

Belz, Anja, Mike White, Josef van Genabith, Deirdre Hogan, and Amanda Stent. 2010. Finding common ground: Towards a surface realisation shared task. In *Proceedings of the 6th International Natural Language Generation Conference (INLG'10)*.

Bohnet, Bernd, Leo Wanner, Simon Mill, and Alicia Burga. 2010. Broad coverage multilingual deep sentence generation with a stochastic multi-level realizer. In *To appear in Proceedings of COLING-2010*, Beijing, China.

Bresnan, Joan. 1996. LFG in an OT setting: Modelling competition and economy. In M. Butt and T. H. King (eds.), *Proceedings of the First LFG Conference*, CSLI Proceedings Online.

Bresnan, Joan. 2000. Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer (eds.), *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford University Press.

Bresnan, Joan, Anna Cueni, Tatiana Nikitina, and Harald Baayen. 2007. Predicting the Dative Alternation. In G. Boume, I. Kraemer, and J. Zwarts (eds.), *Cognitive Foundations of Interpretation*. Amsterdam: Royal Netherlands Academy of Science.

Bresnan, Joan, and Marilyn Ford. 2010. Predicting Syntax: Processing Dative Constructions in American and Australian Varieties of English. *Language* 86(1):186–213.

Cahill, Aoife. 2009. Correlating Human and Automatic Evaluation of a German Surface Realiser. In *Proceedings of the 47th Annual Meeting of the ACL*, pp. 97–100, Suntec, Singapore. Association for Computational Linguistics.

Cahill, Aoife, Martin Forst, and Christian Rohrer. 2007. Stochastic realisation ranking for a free word order language. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, pp. 17–24, Saarbrücken, Germany. DFKI GmbH. Document D-07-01.

Cahill, Aoife, and Arndt Riester. 2009. Incorporating Information Status into Generation Ranking. In *Proceedings of the 47th Annual Meeting of the ACL*, pp. 817–825, Suntec, Singapore. Association for Computational Linguistics.

Crouch, Dick, and Tracy Holloway King. 2006. Semantics via F-Structure Rewriting. In M. Butt and T. H. King (eds.), *Proceedings of the LFG06 Conference*.

Crouch, Richard, Tracy Holloway King, John T. Maxwell III, Stefan Riezler, and Annie Zaenen. 2004. Exploiting F-structure Input for Sentence Condensation. In M. Butt and T. H. King (eds.), *Proceedings of the LFG04 Conference*, University of Canterbury.

Emele, Martin C., Michael Dorna, Anke Lüdeling, Heike Zinsmeister, and Christian Rohrer. 1996. Semantic-based transfer. In *Verbmobil: Foundations of Speech-to-Speech Translation*.

Goldwater, Sharon, and Mark Johnson. 2003. Learning OT constraint rankings using a maximum entropy model. In J. Spenader, A. Eriksson, and Ö. Dahl (eds.), *Proceedings of the Stockholm Workshop on 'Variation within Optimality Theory. April 26-27, 2003 at Stockholm Univ. Sweden*, pp. 111–120.

Graham, Yvette. 2010. *Deep Syntax in Statistical Machine Translation*. PhD thesis, Dublin City University, School of Computing.

Jäger, Gerhard. 2004. Maximum entropy models and stochastic optimality theory. Manuscript, University of Potsdam.

Kaplan, Ronald M., and Jürgen Wedekind. 2000. LFG generation produces context-free languages. In *Proceedings of COLING-2000*, pp. 297–302, Saarbrücken.

Kuhn, Jonas. 2000. Faithfulness violations and bidirectional optimization. In M. Butt and T. H. King (eds.), *Proceedings of the LFG 2000 Conference, Berkeley, CA*, CSLI Proceedings Online, pp. 161–181.

Kuhn, Jonas. 2001. Generation and parsing in Optimality Theoretic syntax – issues in the formalization of OT-LFG. In P. Sells (ed.), *Formal and Empirical Issues in Optimality-theoretic Syntax*, pp. 313–366. Stanford: CSLI Publications.

Kuhn, Jonas. 2003. *Optimality-Theoretic Syntax—A Declarative Approach*. Stanford, CA: CSLI Publications.

Riezler, Stefan, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the ACL*, pp. 271–278, Morristown, NJ, USA. Association for Computational Linguistics.

Riezler, Stefan, and John Maxwell. 2006. Grammatical Machine Translation . In *Proceedings of HLT-NAACL'06*, New York.

Velldal, Erik. 2008. *Empirical Realization Ranking*. PhD thesis, University of Oslo, Department of Informatics.

Wedekind, Jürgen. 1999. Semantic-driven generation with LFG- and PATR-style grammars. *Computational Linguistics* 25:277–281.

Zarrieß, Sina. 2009. Developing German semantics on the basis of parallel LFG grammars. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pp. 10–18, Suntec, Singapore. Association for Computational Linguistics.