Corpus-based Learning in Stochastic OT-LFG – Experiments with a Bidirectional Bootstrapping Approach

Jonas Kuhn

Stanford University[*]
Department of Linguistics

University of Texas at Austin
Department of Linguistics

jonask@mail.utexas.edu

Proceedings of the LFG02 Conference

National Technical University of Athens, Athens

Miriam Butt and Tracy Holloway King (Editors)

2002
CSLI Publications
http://csli-publications.stanford.edu/

[*] The reported work was conducted at Stanford University. The author was supported by a postdoctoral fellowship of the German Academic Exchange Service (DAAD).

Abstract

This paper reports on experiments exploring the application of a Stochastic Optimality-Theoretic approach in the corpus-based learning of some aspects of syntax. Using the Gradual Learning Algorithm, the clausal syntax of German has to be learned from learning instances of clauses extracted from a corpus. The particular focus in the experiments was placed on the usability of a bidirectional approach, where parsing-directed, interpretive optimization is applied to determine the target candidate for a subsequent application of generation-directed, expressive optimization. The results show that a bidirectional bootstrapping approach is only slightly less effective than a fully supervised approach.

1 Introduction

In Optimality Theory (OT), learning of a language amounts to determining the ranking relation over a given set of constraints. Under the target ranking, the observed language data have to be predicted as optimal (most harmonic) among the realization alternatives for the underlying meaning, or input. The fact that one alternative and not another is observed provides indirect negative evidence, which is exploited in learning algorithms (triggering a constraint re-ranking). A robust alternative to the original OT learning algorithm of Tesar and Smolensky (1998) is provided by Boersma (1998), Boersma and Hayes (2001):[1] the Gradual Learning Algorithm (GLA), which assumes a continuous scale for the constraint ranks. With a stochastic component in the determination of the effective constraint ranks, grammars can reflect variation in the training data, while effectively displaying categorical behaviour for most phenomena. This property has been exploited in the analysis of variation in syntax (Bresnan and Deo 2001, Koontz-Garboden 2001, Dingare 2001, Bresnan et al. 2001), based on the OT-LFG framework which uses LFG representations for the candidates, with the f-structures corresponding to the input and (mainly) the c-structure and lexical contribution differing across candidates (Bresnan 2000, Sells 2001b, Kuhn 2001a, forthcoming).

Experimental applications of GLA have so far adopted the idealization that not only the surface form of learning data is known, but the full analysis, including the input (and thus the entire candidate set). With this information, misinterpretations of the evidence for re-rankings are excluded, however a plausible learning approach cannot keep up this idealization. Furthermore, most studies have applied the GLA on a carefully controlled data set, focusing on variation in a small set of phenomena (i.e., keeping other choices fixed by design).

In this paper, I explore the application of GLA for learning clausal syntax, essentially from free corpus data (in the present study from a newspaper corpus of German). The candidate generation grammar is kept highly general, with the only inviolable restrictions being an extended X-bar scheme (in which all positions are

[1] An implementation of the GLA is included in the Praat program by Paul Boersma and David Weenink: `http://fonsg3.let.uva.nl/praat/`

optional). Crucially, I do not assume full syntactic analyses of the learning data as given. I make the weaker, and arguably much more plausible assumption that the learner can use language-independent evidence to narrow down the space of possible semantic representations for an observed form. In the corpus-based learning experiment this narrowing-down is simulated as follows: as training data I use individual clauses (main clauses or subclauses) extracted from a treebank, with a given underlying predicate-argument structure and the argument and modifier phrases pre-bracketed as fixed chunks, as shown in (1).

(1)	[So	streng]	[sind]	[auf	den	Gipfeln]	[die	Sitten	und	die	Gesetze	der	Eitelkeiten]
	So	strict	are	on	the	summits	the	customs	and	the	rules	of	vanities

With the clause boundaries and dependent phrases fixed, experiments with a bootstrapping approach building on a **bidirectional learning** scheme become possible. Under the bidirectionality assumption[2], the same constraint ranking that determines the grammatical form in expressive optimization (based on a fixed underlying meaning) is used in interpretive optimization: for a given string, the most harmonic parsing analysis is taken to be correct. Even though the space of possible interpretations is narrowed down, parsing with the liberal underlying grammar yields an average of more than 16 analyses for short sentences (with four or less “chunks”), so the interpretive optimization is not trivial.

2 OT Syntax background

This paper builds on the OT-LFG framework (Bresnan 1996, 2000, Kuhn forthcoming), in which an Optimality-Theoretic grammar for syntax is formalized based on LFG representations. The OT-LFG architecture is sketched for an example in the diagram in figure 1 and is introduced informally in the following. The small grammar fragment used for this illustration is essentially Bresnan’s OT-LFG reconstruction of Grimshaw (1997) (Bresnan 2000, sec. 2).

A highly general LFG grammar G_{inviol} is assumed that constrains the set of universally possible c-structure/f-structure pairs, i.e., it encodes a basic (extended) X-bar scheme, but is very unrestrictive. In the standard expressive optimization, the set of *candidate structures* is defined as those G_{inviol}-analyses (c-structure/f-structure pairs) which share the semantically interpreted part of the f-structure (the “*input*”). So, there are different potential syntactic realizations of the same meaning to choose from. OT constraints (such as the ones in (2)) are structural descriptions of subparts of a c-structure, an f-structure or of both structures (related through the projection function ϕ). Subparts of the actual candidate structures may violate some of the descriptions/constraints, so the constraint set defines a *constraint violation profile* for each candidate structure.

[2] (Smolensky 1996), for discussion in OT-LFG (outside the learning context) see Lee (2001), Kuhn (2000, 2001b).

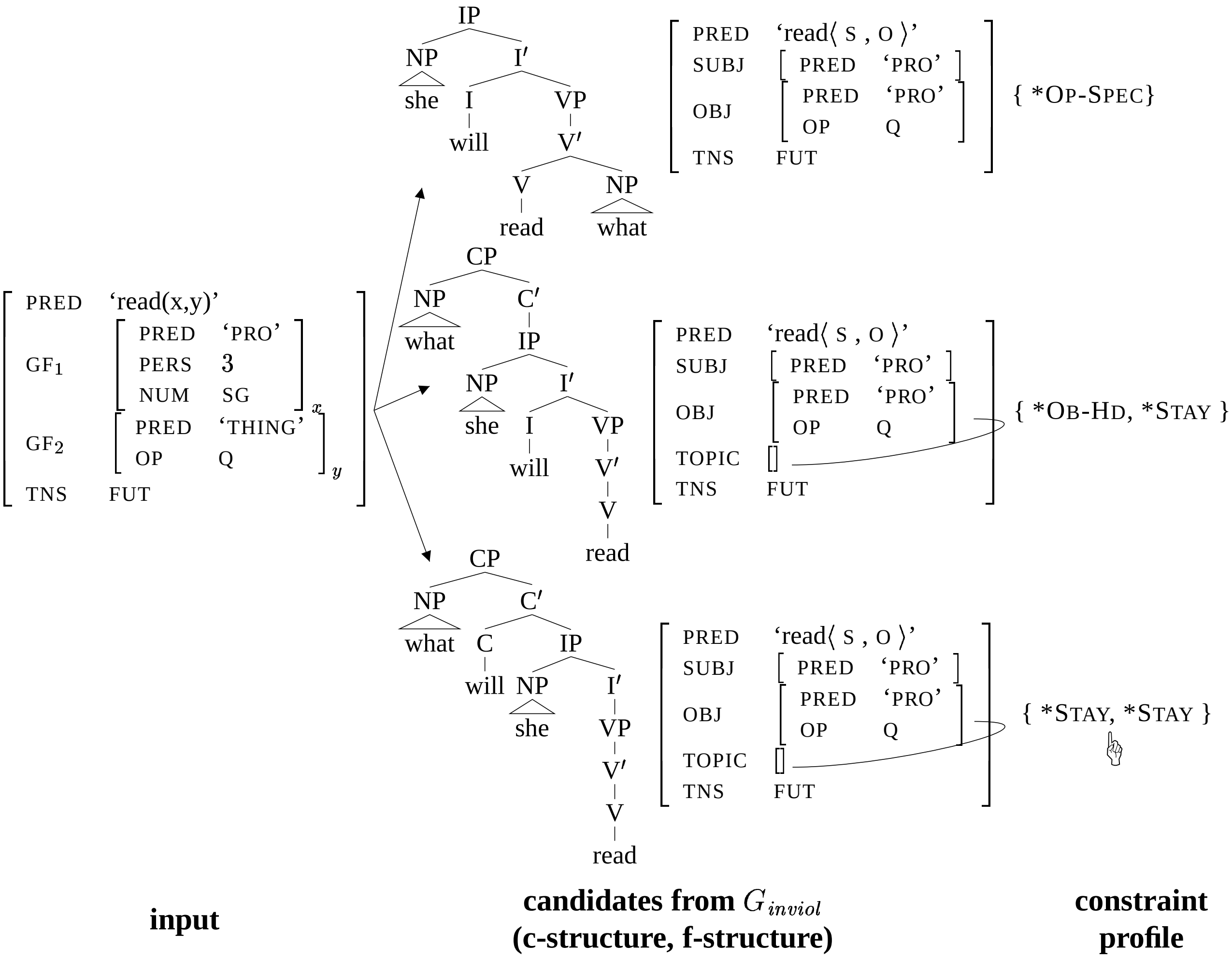

input
candidates from G_{inviol} (c-structure, f-structure)
constraint profile
PRED ‘read(x,y)’
GF1 PRED ‘PRO’ PERS 3 NUM SG x
GF2 PRED ‘THING’ OP Q y
TNS FUT
IP NP I′ she I VP will V′ V NP read what
PRED ‘read⟨ S , O ⟩’ SUBJ [PRED ‘PRO’] OBJ [PRED ‘PRO’ OP Q] TNS FUT
{ *OP-SPEC}
CP NP C′ what IP NP I′ she I VP will V′ V read
PRED ‘read⟨ S , O ⟩’ SUBJ [PRED ‘PRO’] OBJ [PRED ‘PRO’ OP Q] TOPIC [] TNS FUT
{ *OB-HD, *STAY }
CP NP C′ what C IP will NP I′ she VP V′ V read
PRED ‘read⟨ S , O ⟩’ SUBJ [PRED ‘PRO’] OBJ [PRED ‘PRO’ OP Q] TOPIC [] TNS FUT
{ *STAY, *STAY }
☞

Figure 1: A sketch of the OT-LFG architecture (expressive optimization)

(2) a. OP-SPEC (Bresnan 2000)
An operator must be the value of a DF [discourse function] in the f-structure.

b. OB-HD (Bresnan 2000, (21))
Every projected category has a lexically filled [extended, JK] head.

c. STAY (Bresnan 2000, (24))
Categories dominate their extended heads.

Given the language-specific ranking of constraint importance, different structures from the set of candidates arise as optimal in the sense of violating the fewest of the most important constraints (see section 3 for some more discussion). In English (3), it is more important to mark the scope of *wh*-elements overtly than to realize arguments in their canonical position; in a *wh*-in situ language the situation is different: (4). Only the optimal candidate is defined to be a grammatical realization of the underlying part of the f-structure (“input”).

(3) a. **R1:** Op-Spec ≫ Ob-Hd ≫ Stay: *English*

b.

Candidate set:	Op-Spec	Ob-Hd	Stay
[$_{IP}$ she will [$_{VP}$ read what]]	*!		
[$_{CP}$ what [$_{IP}$ she will [$_{VP}$ read]]]		*!	*
☞ [$_{CP}$ what **will** [$_{IP}$ she [$_{VP}$ read]]]			**

(4) a. **R2:** Stay≫ Op-Spec ≫ Ob-Hd: *wh* in situ language

b.

Candidate set:	Stay	Op-Spec	Ob-Hd
☞ [$_{IP}$ “she” “will” [$_{VP}$ “read” “what”]]		*	
[$_{CP}$ “what” [$_{IP}$ “she” “will” [$_{VP}$ “read”]]]	*!		*
[$_{CP}$ “what” **“will”** [$_{IP}$ “she” [$_{VP}$ “read”]]]	*!*		

Interpretive optimization The general architecture of standard expressive optimization is easily adapted to a slightly different formal system (Kuhn forthcoming, ch. 5): if the set of competing candidate structures is not defined by a common semantic representation, but by a common surface string, we get a system of *interpretive optimization*. Rather than choosing from different potential syntactic realizations of a meaning, the OT evaluation now chooses from different syntactic structures (many of which differ in semantic interpretation too) for a given surface string.

This “reverse” formal system has been adapted for a variety of linguistic modeling tasks, in particular for a derivation of the discrepancy between production and comprehension in language acquisition (Smolensky 1996), and in syntax to model word order freezing effects (Lee 2001, Kuhn 2001b). Interpretive optimization may also be assumed in the learning procedure for a standard expressive OT grammar, which will be discussed in section 5.1.

3 Ranking vs. weighting

The previous discussion—like most of the linguistic work in OT—took a central OT assumption for granted: The relative importance of the constraints for a specific language is determined by a *strict ranking*. This means that violating a high-ranking constraint is worse than arbitrarily many violations of some lower-ranking constraint. The ranking scheme is more restrictive than a summation over weighted constraints would be (which one might have chosen as a more general way of computing the joint effect of constraints of different importance, and which is for instance underlying the predecessor of OT, Harmony Grammar).

The OT hypothesis of strict ranking is motivated for the very reason of making the system more restrictive, such that clearly testable typological predictions of the system follow from the assumption of a particular set of constraints. To illustrate this point, let us briefly compare the way predictions are grounded in a ranking scheme and how this compares to a weighting scheme.

If we have a constraint violation profile as in tableau (5) (with the ranking of the constraints open) and we observe candidate A in the data, we know that Constr. 3 must outrank the other two constraints: Constr. 3 ≫ { Constr. 1, Constr. 2 }—else candidate A would be the winner. This kind of configuration is called a *ranking argument*. The fact that candidate B incurs three violations of Constr. 3 and not just one is irrelevant: the only way that B will lose against A is when Constr. 3 is ranked highest.

(5)

Candidate set:	Constr. 1	Constr. 2	Constr. 3
candidate A	*	*	
candidate B			***

Now, if in addition to (5), we observe the A′ and A″ candidate of (6-b) and (6-c) data for the same language, we get an inconsistency: (6-b) and (6-c) are ranking arguments for Constr. 2 ≫ Constr. 3, and Constr. 1 ≫ Constr. 3, respectively.

(6) *Under the ranking hypothesis, (a) is incompatible with (b) and (c)*

(a) Candidate set:	Constr. 3	Constr. 1	Constr. 2
☞ candidate A		*	*
candidate B	***		

(b) Candidate set:	Constr. 1	Constr. 2	Constr. 3
☞ candidate A′			*
candidate B′		*	

(c) Candidate set:	Constr. 1	Constr. 2	Constr. 3
☞ candidate A″			*
candidate B″	*		

So, a small set of clear data is already very informative about an OT account, based on the ranking hypothesis. If we do observe all the data in (6) in a single language, we know that the constraint set assumed was inadequate; maybe an additional constraint or an entirely different set of constraints is needed.

Now, under a *constraint weighting* regime no such clear conclusion about the symbolic part of the theory—the constraints and the candidate representations—can be drawn. The (a) type of data may be compatible with (b), with (c), with both, or none. Examples (7)–(9) illustrate this with different negative weights assumed for the constraints (the winner is defined to be the candidate with the greatest weighted sum over violation marks, e.g., (7-b,A′) wins over (7-b,B′) since $-4 < -3$). In all cases the (a) data are correctly predicted, since $w($Constr. $1) + w($Constr. $2) > 3 \times w($Constr. $3)$. Note however that absolute constraint weights would lead to different rankings in each of the cases, as is suggested by the order of notation (which of course has no technical effect, since we are looking at a weighting system).

(7) *(a) is compatible with (b), but not with (c)*

(a) Candidate set:	Constr. 1	Constr. 3	Constr. 2
	−4	−3	−1
☞−5 cand. A	*		*
−9 cand. B		***	

(b) Candidate set:	Constr. 1	Constr. 3	Constr. 2
	−4	−3	−1
☞−3 cand. A′		*	
−4 cand. B′	*		

not compatible

(c) Candidate set:	Constr. 1	Constr. 3	Constr. 2
	−4	−3	−1
☞−3 cand. A″		*	
−1 cand. B″			*

(8) *(a) is compatible with (b) and (c)*

(a) Candidate set:	Constr. 1	Constr. 2	Constr. 3
	−6	−5	−4
☞−11 cand. A	*	*	
−12 cand. B			***

(b) Candidate set:	Constr. 1	Constr. 2	Constr. 3
	−6	−5	−4
☞−4 cand. A′			*
−6 cand. B′	*		

(c) Candidate set:	Constr. 1	Constr. 2	Constr. 3
	−6	−5	−4
☞−4 cand. A″			*
−5 cand. B″		*	

(9) *(a) is compatible with neither (b) nor (c)*

(a) Candidate set:	Constr. 3	Constr. 1	Constr. 2
	−3	−2	−1
☞−3 cand. A		*	*
−9 cand. B	***		

not compatible

(b) Candidate set:	Constr. 3	Constr. 1	Constr. 2
	−3	−2	−1
☞−3 cand. A′	*		
−2 cand. B′		*	

not compatible

(c) Candidate set:	Constr. 3	Constr. 1	Constr. 2
	−3	−2	−1
☞−3 cand. A″	*		
−1 cand. B″			*

As the example illustrated, the constraint weighting scheme has an undesirable property if we are interested in finding a linguistically motivated set of constraints for predicting a typological spectrum of languages: the effect of picking a particular constraint set is underdetermined—a readjustment of the constraint weights may have the same effect as a modification of the constraint set, i.e., the symbolic part of the theory. This motivates the OT assumption of a constraint ranking regime. The strong interpretation of the OT constraint set assumes that the constraint set reflects innate restrictions on possible grammars (i.e., it formalizes Universal Grammar).

Limitations due to the ranking hypothesis Related to its restrictiveness, the ranking hypothesis has the effect that the phenomenon of optionality or variability of output forms for a single underlying input becomes almost impossible to derive. The strict constraint ranking differentiates between any two candidates with a different constraint profile, predicting all but one candidate to be ungrammatical.[3]

There are different possible ways of overcoming the limitations: one could assume a more fine-grained input representation, distinguishing between cases of optionality; the selection of this input representation itself could be modelled by a contextually controlled process, which may not be fully deterministic. A different modification of the strict OT system is the assumption of a less fixed ranking of the constraints (Anttila 1997, Boersma 1998, Boersma and Hayes 2001). The stochastic OT system proposed by Boersma will be discussed in more detail in the next section. Yet another option might be to assume a weighting scheme where the weights are typically widely separated, so the emerging behavior is almost that of a ranking scheme.

It is fairly difficult to find independent criteria for deciding between the various choices in the architecture of such a modified OT system: applying the systems for a non-trivial learning task, as is attempted in this paper, is one way of assessing their adequacy (although this alone may not lead to a conclusive answer).

4 Learning

The learning procedures that have been proposed for Optimality Theory are essentially *error-driven*. This means that during learning, a hypothetical constraint ranking is applied to the learning data. Under a simplifying assumption (which will be challenged in section 5.1), the learner has access to the underlying input representation for an observed piece of learning data; with the candidate set being defined in terms of G_{inviol} and the input, the learner has thus access to the full set of candidates. The learner will then need some monitoring ability, in order to be able to compare its own predictions of the output/winner, based on the hypothetical constraint ranking, with the output in the actual data. Whenever there is a mismatch, this is evidence that the hypothetical ranking cannot be (fully) correct.

For instance, in (10) the hypothetical ranking CONSTR. $1 \gg$ CONSTR. $2 \gg \ldots \gg$ CONSTR. 5 would predict candidate A to be the winner. But the observed output structure is candidate B. Hence, the assumed ranking must have been incorrect: CONSTR. 3

[3] Of course more than one candidate can have the same constraint profile, but with a realistic constraint set, this is no modelling option for most cases of optionality.

should outrank Constr. 1.

(10) *Detecting an error in the learner's system*

Candidate set:	Constr. 1	Constr. 2	Constr. 3	Constr. 4	Constr. 5
candidate A		*	*		
observed: candidate B	*!	*			*

In the Constraint Demotion Algorithm (Tesar and Smolensky 1998), this type of ranking argument is exploited to make conservative modifications of the ranking, which guarantee that learning will converge (on noise-free data). Constraints violated by both the predicted winner (A) and the observed output (B) and constraints violated by neither of the two are ignored in a learning step. Of the remaining constraints, the ones violated by observed output are demoted just below highest-ranking constraint violated by putative winner. So Constr. 1 is demoted just below Constr. 3:

(11) *Constraint demotion*

Candidate set:	Constr. 1	Constr. 2	Constr. 3	Constr. 4	Constr. 5
candidate A		*	*		
observed: candidate B	*	*			*

(12) *Constraint ranking after learning step*

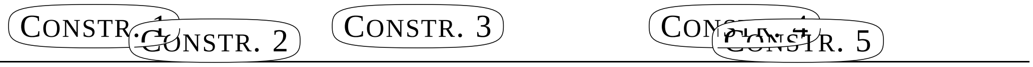
Candidate set:	Constr. 2	Constr. 3	Constr. 1	Constr. 4	Constr. 5
candidate A	*	*!			
observed: candidate B	*		*		*

The Gradual Learning Algorithm (GLA) Since the Constraint Demotion Algorithm was developed for the strict OT ranking architectire, it cannot be used to learn from data displaying optionality/variation. Also, the algorithm is not robust; i.e., a single instance of data incompatible with the target ranking may corrupt the intermediate ranking in a way from which the learner cannot recover. Boersma (1998), Boersma and Hayes (2001) propose an alternative learning algorithm, the Gradual Learning Algorithm (GLA), based on a modified ranking architecture, which is robust and can deal with optionality.

In the modified architecture—stochastic OT—the constraint ranking is no longer discrete, but the constraints are ranked on a continuous scale: the rank or strength of a constraint is represented by a numerical value. (However, we still have a ranking and not a weighting, i.e., just the relative strengths of constraints are relevant; there is no summation over the values of the violated constraints.) As the candidates in a tableau are evaluated, some random noise with a normal distribution is added to the constraint strength. This can have the effect of reversing the effective order of the constraint and thus leads to a variable behavior of the system.

Diagram (13) is a schematic illustration of a set of constraints ranked on the continuous scale, with strength decreasing from left to right. When the constraint strengths (i.e., the means of the normal distribution) are sufficiently far apart—as for CONSTR. 3 vs. CONSTR. 4—a reversal will effectively never happen, so we have a categorical effect like with a discrete ranking. For constraints with a similar strength (like CONSTR. 4 and CONSTR. 5), we will however find both orders, depending on the noise at evaluation time.

(13)



In the GLA, designed for stochastic OT, a learning step (triggered by an observed error like in the Constraint Demotion Algorithm) does not lead to a readical change in the constraint ranks. Rather, a slight adjustment of the constraint ranks is made, promoting the constraints violated by the erroneous winner, and demoting the constraints of the observed output:

(14) *Promotion/demotion in the GLA*

Candidate set:	CONSTR. 1	CONSTR. 2	CONSTR. 3	CONSTR. 4	CONSTR. 5
candidate A		*	*		
observed: candidate B	*	*			*
	→		←		→

Data types occurring with sufficient frequency will cause a repeated demotion/promotion, so a quasi-categorical separation of the constraint strengths can result; noise in the data will have only a temporary effect. In variability phenonema, opposing tendencies of constraint demotion/promotion will ultimately balance out in a way that reflects the frequencies in the data (assuming a large enough sample is presented to the learner).

As applications of the GLA in phonology and syntax (see the citations in section 1) have shown, the algorithm is able to adjust the constraint strengths for the linguistic constraint sets posited in these studies in an appropriate way: the behavior of the stochastic model indeed replicates the frequency distribution of the data types in the learning data.[4] However, so far GLA applications have focused on relatively small, clear-cut grammar fragments.

5 Experiments

The experiments reported in this paper address the following questions: (i) Can GLA be used for an exploratory analysis of a more complex cluster of interacting phenomena? (ii) What is the amount of target information required to control the error-based learning scheme?

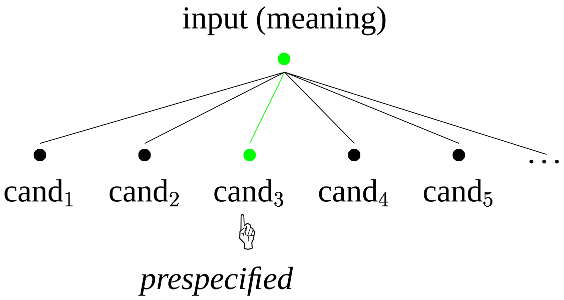
Methodologically, the idea was to start out with a certain set of linguistically well-understood constraints, and to add further constraints in order to explore interactions. The set of phenomena to be chosen for this investigation was supposed to display variation, but at the same time clearly obey certain language-specific principles. Under these criteria, the clausal syntax of German is a well-suited target for learning: the system is confronted with a high degree of word order variation in the relative order of argument phrases in the *Mittelfeld* (the area following the finite verb in matrix clauses), but the verb position in the various clause types is fixed and has to be learned as categorical facts. The exact way of representing the training data from a corpus was motivated by considerations concerning the “degree of supervision” in learning (question (ii)), which is discussed in the following subsection.

5.1 Target information in learning

How much information should be provided to the learner with the learning data? Previous studies of learning in OT—both for the constraint demotion algorithm and for the GLA—have assumed the following idealization: the learner is presented with the full candidate set (which is constructable from the exact input), plus the exact target output candidate (compare the diagram in (15)). This means that an error in the predictions of the learner’s system can be very reliably detected—if any other candidate than the target output is more harmonic, we have an error.

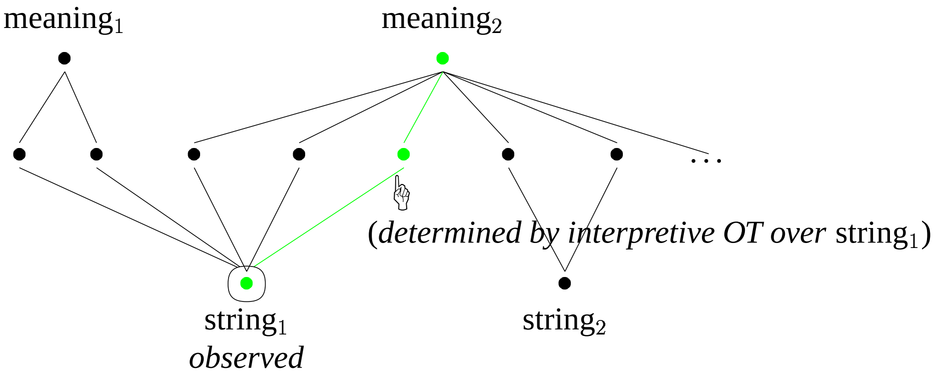
[4] Keller and Asudeh (2001) observe that for certain constraint sets that have been assumed in the linguistic literature, the GLA does not converge; however this may indicate that the assumed constraints are insufficient for an adequate description of the data.

(15) *Full target annotation (schematic)*



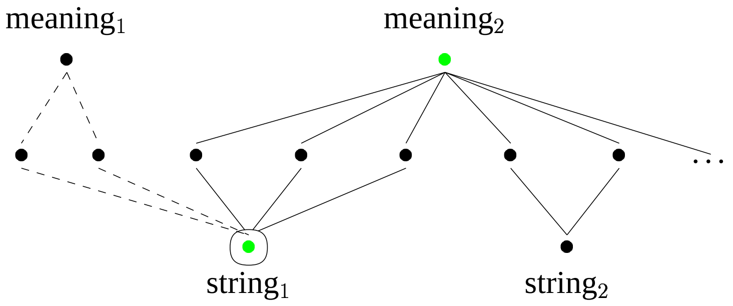
Of course, the only direct observation that a human learner has access to is the surface form (of utterances made by adult speakers). There may be many different underlying inputs for a given surface form, and even for the same combination of input and surface string, there may be differences in the syntactic analysis. In theoretical OT work, a process of *robust interpretive parsing* is assumed, which the learner applies to "guess" what the underlying input for an observed string is (Tesar and Smolensky 1998). The current constraint ranking is simply applied on the set of candidates defined by a common surface string (parsing-based or interpretive optimization). Based on the underlying input determined in this way, the standard generation-based or expressive optimization is applied as the basis for the actual learning (compare (16)).

(16) *Determining the target for expressive optimizarion by interpretive optimization*



Hence, the mentioned idealization in the presentation of the target structure is not hard-wired into the OT architecture. A bidirectional of optimization (robust interpretive parsing, plus expressive optimization) works without this assumption. In the long run, one may hope that corpus-based learning experiments can apply the general bidirectional strategy. However, based exclusively on linguistic material, a corpus-based learner has a considerable disadvantage: the human learner can exploit semantic information and background knowledge, and this way the choices in interpretive parsing are often narrowed down considerably. In the present experiments, I tried to simulate this effect by providing the full predicate-argument structure (i.e., the full underlying input) for the learning instances. This still leaves open which of the syntactic analyses for the observed string is the right target winner.

(17) *Narrowed down set of choices in interpretive optimization*



5.2 Experimental set-up

The training data were extracted from the TIGER treebank, a syntactically annotated newspaper corpus of German (cf. Brants et al. (2002), Zinsmeister et al. (2002)). The treebank includes full categorial and functional annotations, but this information was of course only partially exploited for training data (as far as justified by non-syntactic information available to the human learner).

The data was split up into single clauses, i.e., either matrix clauses or embedded clauses (presented as separate training instances). Since the focus was on the learning of clausal syntax, embedded argument/modifier phrases (NPs, PPs, etc.), were pre-bracketed, and their grammatical functions were provided. No syntactic information was provided about verbal constituents, i.e., verbs and auxiliaries were left as separate, unconnected units.

For example, sentence (18) would give rise to two training instances (19)—one for the matrix clause, including a single "chunk" for the embedded complement clause, and one for the internal structure of the complement clause.

(18) Der Vorstand der Firma hat gefordert, daß der Geschäftsführer entlassen
the board of the company has demanded that the managing director laid off
wird.
is

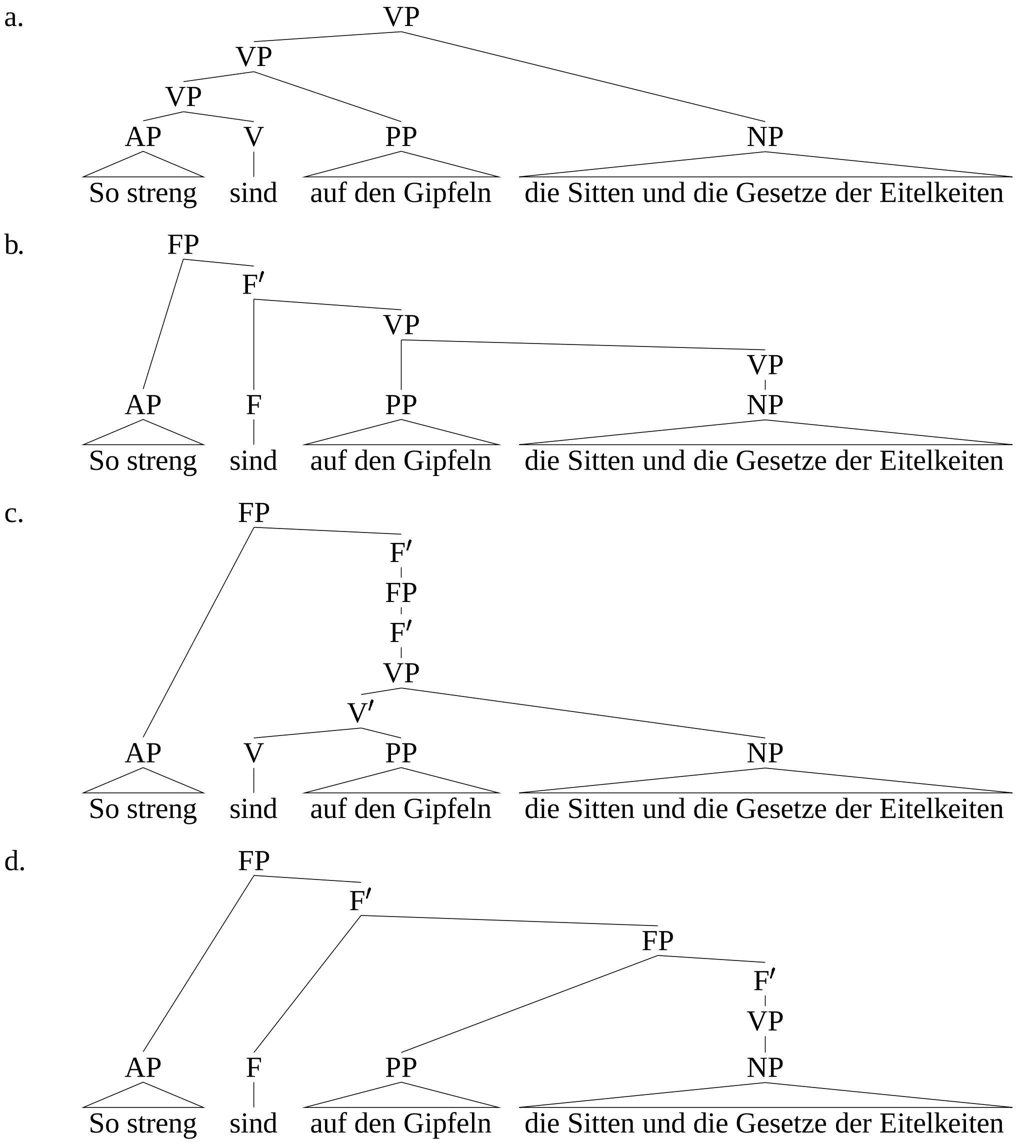
(19) a. [Der Vorstand der Firma] hat gefordert, [daß . . .]

b. daß [der Geschäftsführer] entlassen wird

The candidate analyses The set of candidates was generated by a highly underrestricted LFG grammar (G_{inviol}), approximating the OT hypothesis that all universally possible structures should be included in this set. Reflecting inviolable principles, an extended X-bar scheme is encoded in the LFG grammar; the scheme is very general however, all positions are optional, functional projections (IP, CP) can be freely filled with verbs, auxiliaries, complementizers. The grammar was written and applied with Xerox Linguistic Environment (XLE).[5] As an illustration of the broad range of analyses licensed by the underlying grammar, consider the sample structures in (21) for sentence (20).

(20) [So streng] [sind] [auf den Gipfeln] [die Sitten und die Gesetze der Eitelkeiten]
So strict are on the summits the customs and the rules of vanities

(21)



The OT constraints The constraints were also encoded using XLE (compare Frank et al. (2001)). The core constraints adopted were inspired by OT accounts of clausal

[5] For technical reasons, a generation-based application of the grammar was simulated by parsing all permutations of the string. In future experiments, it should be possible to use the XLE generator.

syntax (Grimshaw 1997, Sells 2001a); further constraints were added to ensure distinguishability of candidates. A total of about 90 constraints was used—based on X-bar configurations, precedence relations of grammatical functions/NP types (pronominal vs. full), etc.

Due to computation-intensive preprocessing routines for the learning data, required after each change in the assumed constraint set, the learning experiments were only performed on small training sets. The reported results are from a specific sequence of experiments based on 195 training sentences.

5.3 Learning schemes

The corpus-based learning was performed with the GLA (using a simple Prolog implementation) in generation-based optimization. As discussed in sec. 4, the GLA is an error-based learning algorithm, i.e., at each state, the learner applies its present, hypothetical ranking. If the predicted winner matches the target output, no adjustment is necessary; if a different output is the target, the constraints violated only by the predicted winner have to be promoted, while those violated only by the target output are demoted.

As discussed in sec. 5.1, a realistic approach should compute the target winner based on a bidirectional approach. In order to test the feasibility of such an account—within the limits of the assumptions discussed above—three different learning schemes were compared in the experiment:

1. The “fully supervised” scheme:
   The exact target structure for the training clauses was manually annotated (based on the standard analysis of German clause structure).

2. The “string-as-target” scheme:
   No manual annotation was made; all candidates with the right word order count as target winners (no interpretive optimization is performed). Only predicted winners with an incorrect surface order count as errors—i.e., constraints violated by *any of the target winners* (and not the predicted winner) are demoted.

3. The bidirectional optimization (or “bootstrapping”) scheme:
   The current ranking is used to determine a target winner among parsing alternatives for the observed string. All other candidates (possibly with correct surface order) count as errors.[6]

[6] For the bidirectional scheme, two variants were compared: one, in which the same effective ranking—i.e., the ranking after addition of noise—was used in generation and parsing; and another one, in which the initial parsing-based optimization was sampled several times (leading to different effective rankings), in order to determine a larger set of target winners. The evaluation showed that both variants lead to a very similar behavior.

5.4 Results

Evaluation schemes It is not straightforward how to best evaluate the performance of a generation-based optimization system. Demanding that the word string predicted for an unseen underlying predicate-argument structure be an exact match of the actual string in the corpus would be too strict, since there are many cases of real optionality: even in the concrete given context, several orderings are perfectly natural. Instead of evaluating how often the exact string in the corpus is predicted for unseen generation tasks, the main evaluation measure is based on a manual annotation of the acceptable permutations for a set of 100 evaluation sentences, which had not been presented as training data. All natural-sounding permutations in the given context were annotated as possible generation alternatives. No inter-subject comparsion of the annotations was made, so the raw percentage numbers for the various learning schemes should be treated with some caution. The focus of the experiments was on a *comparison* of the different schemes.

Besides this main evaluation measure, a variation of the bidirectional optimization technique was applied: the ranking that the learner came up with (through generation-based learning, possibly with a parsing-based determination of the target winner) is used in a disambiguation task. For sentences with ambiguous case marking on the argument phrases, a theory of word order preferences predicts how likely the individual readings are (compare the discussion of word order freezing in bidirectional OT in Kuhn (2001b), Lee (2001)). A corpus example of such an ambiguous case marking is shown in (22): both bracketed NPs can be either nominative or accusative. 50 such unseen examples from the corpus were used for the second evaluation measure, counting how often the intended reading was matched by the system's prediction.

(22)	daß	[die	Bundesregierung]		[die	militärische	Zusammenarbeit]	wiederbelebt	hat
	that	the	federal	government	the	military	cooperation	revitalized	has

Results The evaluation results (for a specific series of experiments) are shown in (23). The left-most column shows the results for the initial ranking (with all constraints ranked the same).

(23) a. *Percent acceptable orderings on unseen data*

initial ranking	"string-as-target"	bidirectional	"supervised"
34%	66%	87%	90%

b. *Disambiguation of unseen parsing ambiguities*

initial ranking	"string-as-target"	bidirectional	"supervised"
54%	76%	84%	83%

Note that in (23a), the bidirectional approach leads to a significant improvement over the "string-as-target" scheme. For the disambiguation task (23b), the bidirectional scheme is as good as the supervised approach.[7] So both measures indicate that the bidirectional bootstrapping approach is very promising.

6 Discussion

While the question about the usefulness of bidirectional optimization in learning can be answered positively, it is not entirely clear what conclusions can be drawn for the other question: can the GLA be used straightforwardly in an exploratory analysis with a large number of constraints? The large set of constraints seems to make the analysis of linguistic effects somewhat opaque. However, this may be due to a lack of analytical tools.

As I discussed in Kuhn (2002), there are certain cases in which the GLA is not able to deal with conflicting (statistical) ranking arguments. It is possible that the data sets contained such cases. A small experiment using a weighting-based model on the training data from the fully supervised scheme indicated that a better fit on the training data is possible (in this experiment, I used the log-linear model that Johnson et al. (1999) developed for disambiguation of parses with a large-scale LFG grammar[8]).

For deciding what is an adequate linguistically restricted learning model to deal with a larger number of interacting phenomena, further experiments are required. The learning instances should be kept more controlled, without having to move away from the use of real corpus data. A promising approach might be to use a (slightly relaxed) classical large-coverage grammar to produce the learning material.

[7] The fact that it is even slightly better may be an effect of the small size of the training data; it was easier for the bidirectional approach to come up with (potentially incorrect) generalizations over the data, whereas the supervised approach was confronted with the linguistically motivated target annotations, for which there may not have been enough support in the data.

[8] I would like to thank Mark Johnson for providing the learning code.

References

Anttila, Arto. 1997. *Variation in Finnish Phonology and Morphology*. PhD thesis, Stanford University.

Boersma, Paul. 1998. *Functional Phonology. Formalizing the interactions between articulatory and perceptual drives*. PhD thesis, University of Amsterdam.

Boersma, Paul, and Bruce Hayes. 2001. Empirical tests of the gradual learning algorithm. *Linguistic Inquiry* 32:45–86.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.

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 Joost Dekkers, Frank van der Leeuw, and Jeroen van de Weijer (eds.), *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford University Press.

Bresnan, Joan, and Ashwini Deo. 2001. Grammatical constraints on variation: ‘be’ in the Survey of English Dialects and (Stochastic) Optimality Theory. Ms., Stanford University.

Bresnan, Joan, Shipra Dingare, and Christopher Manning. 2001. Soft constraints mirror hard constraints: Voice and person in English and Lummi. In M. Butt and T. H. King (eds.), *Proceedings of the LFG 01 Conference*. CSLI Publications.

Dingare, Shipra. 2001. The effect of feature hierarchies on frequencies of passivization in English. Master’s thesis, Stanford University.

Frank, Anette, Tracy H. King, Jonas Kuhn, and John Maxwell. 2001. Optimality Theory style constraint ranking in large-scale LFG grammars. In Peter Sells (ed.), *Formal and Empirical Issues in Optimality-theoretic Syntax*, pp. 367–397. Stanford: CSLI Publications.

Grimshaw, Jane. 1997. Projection, heads, and optimality. *Linguistic Inquiry* 28:373–422.

Johnson, Mark, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic “unification-based” grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL’99), College Park, MD*, pp. 535–541.

Keller, Frank, and Ash Asudeh. 2001. Probabilistic learning algorithms and Optimality Theory. Ms., Saarbrücken, Stanford University.

Koontz-Garboden, Andrew. 2001. A stochastic OT approach to word order variation in Korlai Portuguese. paper presented at the 37th annual meeting of the Chicago Linguistic Society, Chicago, IL, April 20, 2001.

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. 2001a. *Formal and Computational Aspects of Optimality-theoretic Syntax*. PhD thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

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

Kuhn, Jonas. 2002. Extended constraint ranking models for frequency-sensitive accounts of syntax. Slides for a presentation at the Workshop *Quantitative Investigations in Theoretical Linguistics* (QITL), 3-5 October 2002, Osnabrück, Germany.

Kuhn, Jonas. forthcoming. *Optimality-Theoretic Syntax: a Declarative Approach*. Stanford, CA: CSLI Publications.

Lee, Hanjung. 2001. Markedness and word order freezing. In Peter Sells (ed.), *Formal and Empirical Issues in Optimality-theoretic Syntax*, pp. 63–128. Stanford: CSLI Publications.

Sells, Peter. 2001a. *Alignment Constraints in Swedish Clausal Syntax*. Stanford: CSLI Publications.

Sells, Peter (ed.). 2001b. *Formal and Empirical Issues in Optimality-theoretic Syntax*. Stanford: CSLI Publications.

Smolensky, Paul. 1996. On the comprehension/production dilemma in child language. *Linguistic Inquiry* 17:720–731.

Tesar, Bruce B., and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29:229–268.

Zinsmeister, Heike, Jonas Kuhn, and Stefanie Dipper. 2002. Utilizing LFG parses for treebank annotation. In *LFG 2002, Athens*.

Jonas Kuhn
jonask@mail.utexas.edu
Department of Linguistics
1 University Station, B5100
University of Texas at Austin
Austin, TX 78712-1196
USA