Parsing unrestricted German text with defeasible constraints*

Kilian A. Foth Michael Daum Wolfgang Menzel

Natural Language Systems Group
Hamburg University
D-22527 Hamburg
Germany
{foth,micha,menzel}@nats.informatik.uni-hamburg.de

Abstract. We present a parser for German that achieves a competitive accuracy on unrestricted input while maintaining a coverage of 100%. By writing well-formedness rules as declarative, defeasible constraints that integrate different sources of linguistic knowledge, very high robustness is achieved against all sorts of extragrammatical constructions.

1 Introduction

Although most linguists agree that natural language to a large degree follows well-defined rules, it has proven exceedingly difficult to actually define these rules well enough that a computer could reliably assign the intuitively correct structure to natural language input. Therefore, almost all current approaches to parsing constitute compromises of some kind.

1. Often the goal of full syntactic analysis is abandoned in favour of various kinds of *shallow parsing*, which is all that is needed for many applications. Instead of a full syntax tree, e.g., only the boundaries of major constituents [2] or topological fields [3] are computed.
2. Instead of casting the language description into linguistically motivated rules, a probability model is induced automatically from a large amount of past utterances, which is then used to maximize the similarity to previously seen structures [4]. This approach is robust and efficient, but relies on a large amount of existing data, and the resulting model is difficult to comprehend and extend.
3. Formalisms that do perform deep analysis by following explicit rules typically have to be restricted to particular subsets of linguistic constructions or to particular domains. Also, their robustness and coverage (the performance for ungrammatical and extragrammatical input, respectively) is often rather low on realistic data.

* This is an extended version of a paper published in the proceedings of the 7. Konferenz zur Verarbeitung natürlicher Sprache, Vienna 2004 [1]

We present a parsing system that tries to avoid all three compromises to the extent possible at the current time. Instead of a derivation grammar, we employ a declarative formalism in which well-formedness conditions are expressed as explicit constraints (which take the form of logical formulas) on word-to-word dependency structures. Since there is no limit to the complexity of these formulas, *every* conceivable well-formedness condition can be expressed as a constraint. Although the formal power of this approach would theoretically make the parsing problem intractable, we have found that approximative solution methods yield good results in practice.

All grammar rules are ordered by a preference measure that distinguishes less important rules (e.g., rules of style) from more important fundamental grammar rules. This not only ensures robust behaviour against all kinds of extragrammatical constructions, but also allows us to integrate the information from external shallow parsers, such as a part-of-speech tagger, without becoming dependent on their results.

The output of the system consists of labelled dependency trees rather than constituent-based structures. Although this is mainly a consequence of casting the parsing problem in the form of a constraint optimization problem, it also has benefits for the processing of languages like German with a relatively free word order.

2 WCDG: Constraints on dependency structures

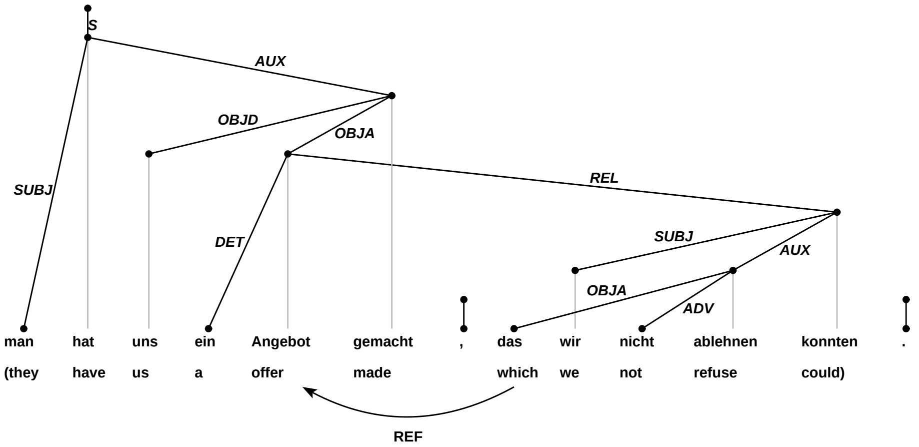
S
AUX
OBJD
OBJA
REL
SUBJ
DET
SUBJ
AUX
OBJA
ADV
man hat uns ein Angebot gemacht , das wir nicht ablehnen konnten .
(they have us a offer made which we not refuse could)
REF

Fig. 1. Dependency analysis of a German sentence.

Weighted constraint dependency grammar (WCDG) [5] is an extension of the CDG formalism first described by [6]. It captures the structure of natural language as a set of labelled subordinations: each word is either subordinated to exactly one other word or considered the root of the syntax tree (also called a NIL subordination). See Figure 1 for an example. Each subordination is annotated with one of a fixed set of labels such as ‘subject’, ‘direct object’ or ‘indirect object’, but no larger groups of words carry labels. This means that there is no direct equivalent to the constituents postulated by phrase structure grammar.

Dependency structures are encoded as values of a set of variables. In this respect the approach closely corresponds to customary constraint satisfaction techniques [7], as they have been successfully applied to a great number of complex problem solving tasks, e.g. in planning and scheduling. For any given sentence to be parsed there is a fixed number of variables corresponding to its individual word forms. Each variable takes as a value a pair consisting of an attachment point for the corresponding word form and the label with which it is subordinated. Accordingly the resulting search space is finite and highly regular. The task of the parser is then to select a set of subordinations that satisfies all constraints of the grammar. With this view on constraint satisfaction the CDG approach differs considerably from constraint-based unification grammars (like HPSG [8]): these formalisms only consider values of high structural complexity for a single variable (namely for the sentence) which, however, may contain embedded variables that receive their values by unifying feature structures.

Since CDG models no constituents, it has no generative rules along the lines of ‘S → NP VP’; these are often problematic for languages with free or semi-free word order since they mingle dominance principles with linear precedence rules. Lacking a context-free backbone, constraints can only refer to the position, reading and lexical features of the concerned word forms, as well as features of neighbouring dependency edges. Only passive checks on the compatibility of certain features are carried out and no update of ambiguous feature assignments according to the syntactic context is possible. Therefore morpho-syntactic ambiguity has to be encoded completely as alternative lexical readings, and the selection between them needs to be carried out by the general disambiguation facilities of the underlying constraint satisfaction procedure.[1] Again this sets the CDG approach apart from unification-based formalisms, where in addition to the boolean result of a constraint application also the resulting feature description of the linguistic sign is obtained. Going without the possibility for feature updates, however, has the advantage that additional constraint-solving techniques become readily available.

3 Parsing as constraint satisfaction

Constraints license subordination possibilities, i.e. value assignments to the variables of the constraint satisfaction problem. Every subordination that is not forbidden by any constraint is considered valid. In standard constraint solving systems, constraints are directly specified for particular subsets of variables. This approach is not feasible for natural language parsing, since here variables have

[1] While often glossed over, this task is quite difficult in languages with a rich morphology; the average German adjective form has 10 morphosyntactic variants that are relevant for agreement.

to be mapped to word forms which only become available at run time. Instead, CDG formulates all-quantified constraints that always apply to all variables or variable tuples in a problem; therefore, relevance conditions have to be included into each CDG constraint, which thus take the general form of an implication. Whenever a constraint fails, it pinpoints the offending words as if it were specific to the particular variables; in fact, an all-quantified constraint can fail multiple times for different words in the same sentence.

As an example of a constraint, consider the rule that normal nouns of German require a determiner of some kind, either an article or a nominal modifier, unless they are mass nouns. This rule can be formulated as a constraint as follows:[2]

```
{X:SYN} : 'missing determiner' : 0.2 :
  X↓cat = NN
  ->
  exists(X↓mass_noun) |
  has(X↓id, DET) |
  has(X↓id, GMOD);
```

It states that for each subordination on the syntax level (SYN), a word with the category 'normal noun' (NN) must either bear the feature 'mass noun' or be modified by a determiner (label DET) or a genitive modifier (label GMOD).

Common constraint satisfaction techniques can now be applied to obtain a structural description for the sentence which obeys all the constraints of the grammar. Among the available approaches are

- constructive algorithms, which build a variable assignment by successively extending partial solutions until a globally satisfactory one has been found,
- pruning procedures, which successively remove values for the domains if they cannot be part of any solution, and
- repair-based techniques, which try to improve a solution by transforming it into another (hopefully better) one guided by observed constraint violations in the current value assignment.

The formal power of constraints can be chosen depending on the available solution procedures. While eliminative pruning procedures and methods for constructive solution generation usually require the restriction of constraints to at most binary ones for efficiency reasons, transformation-based solution methods also allow using constraints for an arbitrary number of variables (i.e. for existence conditions).

In particular for large problem instances that do not allow a complete search to be performed, good success has been achieved with transformation-based heuristic solution methods [9, 10] that approximate the acceptability model defined by the constraints. Consider the example parsing run of the utterance 'Konkursgerüchte drücken den Kurs der Amazon-Aktie' (*rumours of bankruptcy*

[2] This constraint is considerably simplified for exposition purposes, since the rule actually has many systematic exceptions.

lower Amazon's stock price) in Figure 2. A structure is first composed by selecting five possible labelled word-to-word subordinations individually, and errors in the structure are successively removed by changing those subordinations where important constraints fail. In the first step, both the agreement error and the unusual position of the subject following its object are remedied by reinterpreting it as a genitive attribute. In the second step the unusual 'ethical dative' is reinterpreted as the more probable subject, which simultaneously fulfills the corresponding valency constraint of the finite verb.

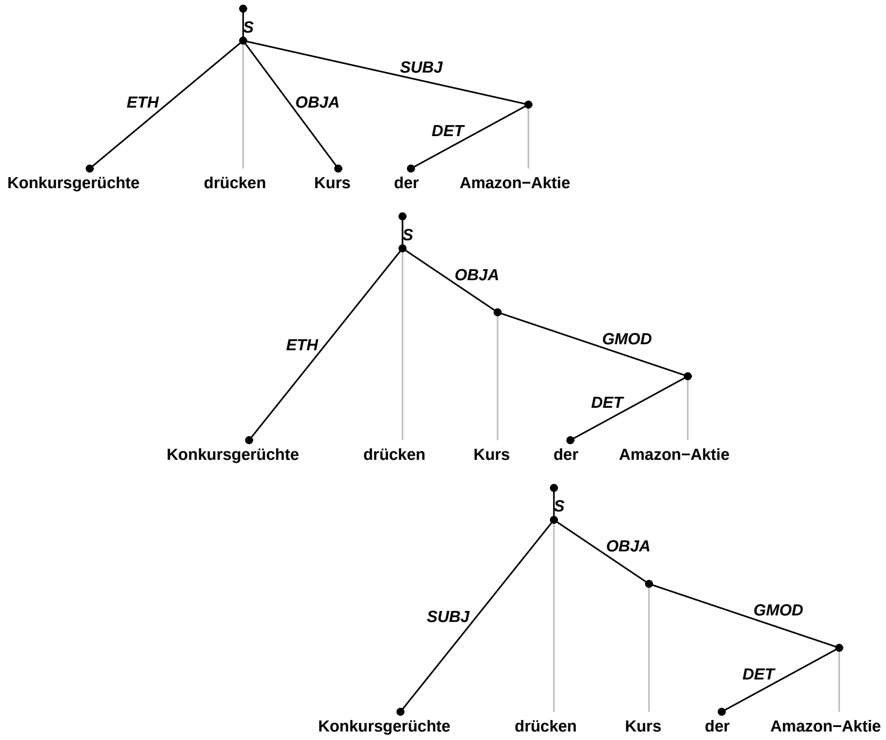

Fig. 2. Successive stages of a transformation-based parsing run.

In a large-scale grammar with many defeasible constraints, different analyses of the same sentence usually carry at least slightly different scores, so that full disambiguation is achieved. If different analyses happen to have the same score, the one that the transformation process finds first is usually selected as the output of the parser[3]. By default, analysis terminates when it has repaired, or

[3] However there is no problem of collecting different parses of the same score the parser happens to find.

failed to repair, all important constraint violations (those whose score lies below a configurable threshold).

4 Weighted constraints

An important feature of the WCDG formalism is the availability of a weighting scheme for constraints. Each constraint bears a score between `0.0` and `1.0` that indicates its importance relative to other constraints. The acceptability of an analysis is defined as the product of the scores of all instances of constraints that it violates.[4] This means that constraints with a score of `0.0` must be satisfied if at all possible, since violating them would yield the worst possible score.[5] Note that the score of the determiner constraint is `0.2`, which means that missing determiners are considered wrong but not totally unacceptable. In fact, many other constraints are considered more important than the determiner rule.

Scores in WCDG serve a range of different purposes. First of all they can be used to model conflicting regularities within the grammar which frequently surface in natural language as all kinds of preferences for linear orderings, attachment points or morpho-syntactic readings. Without the possibility to acknowledge graded phenomena, such information must be ignored completely. This, however, usually prevents binary valued logics from fully disambiguating a parsing problem. Therein the mechanism of weighted constraints shows some similarity to stochastic approaches to natural language parsing. In both cases, parsing amounts to determining the optimal structural description given a particular cost function. Constraint scores also help to guide the parser towards the optimal solution: a transformation-based solution method, for instance, will change those parts of an analysis with the most important constraint failures, and it will choose those alternatives that remedy the failure.

Finally, scores allow the parser to deal with extragrammatical input in a robust way. Since the scores to a large degree mirror the linguistic intuition of a grammar writer about what kind of deviations are more acceptable than others, even in case of constraint violations the resulting structure remains a (partially) meaningful one and the observed constraint violations can even be used as diagnostic information about the kind of error encountered [11].

By assigning values other than `0.0` to all rules that might conceivably be broken, a prescriptive grammar can easily be converted to one that is robust against all kinds of language errors, while still preferring conforming over deviant structures wherever possible. In fact, a coverage of 100% can be guaranteed with an appropriately written grammar. This is achieved by giving those constraints which could forbid *all* structures for a given utterance a higher value.

One easy way of ensuring this property is to allow surplus words to form the roots of isolated subtrees. This is possible because WCDG by itself does

[4] Note that due to the multiplication operator, higher numerical scores correspond to lower linguistic importance.

[5] However, the parser will still return such a structure if no analysis with a positive score can be found.

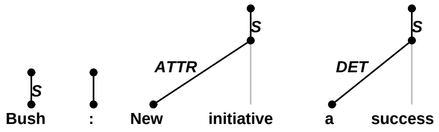

Fig. 3. Analysis of an elliptical utterance.

not enforce a unique NIL subordination. Note that in some cases a set of fragments actually is the most faithful rendition that is possible in a dependency formalism. Consider the typical elliptical headline *"Bush: New initiative a success"*, where the verbs "said" and "is" have been elided. This should actually be modelled as a forest of three tree fragments (cf. Figure 3). Assigning a heavy penalty to fragments that are not finite verbs ensures that they will be avoided unless absolutely necessary. *Partial parsing* can thus be achieved simply by not constraining the number of NIL subordinations.

The selection of real-valued weights for all constraints of a grammar is a difficult problem because of the virtually unlimited number of possibilities. A score for a newly written constraint can be calculated by counting how often it holds and fails on an annotated corpus, or by measuring the overall performance when different weights are assigned to it. The latter method usually shows very little variation; better results are achieved when only sentences that actually contain the described phenomenon are analysed. In general the exact score of a rule is less important than the relative score of two rules that might interact [12].

5 Modelling natural language through declarative constraints

Describing a natural language by writing a constraint grammar amounts to explicitly stating its recurring properties. Each constraint is a logical proposition that describes a particular aspect of well-formed syntax trees, such as 'subject and verb agree in number' or 'subjects precede their verbs'. A parser can then compute the correct analysis for given input by assigning functions to all words so that these constraints are obeyed. The main problem is usually that the same utterance often allows for different analyses that satisfy all the rules of the grammar.

Graded constraints provide a way for a WCDG to counteract this pervasive ambiguity: of two constraints with different scores, the one with the (numerically) higher score will be violated in preference to the other one. In particular, *soft* constraints will be violated in preference to any *hard* constraint (one with the score 0). This allows the grammar developer to formulate multiple linguistic criteria for interpretation, even those that can contradict each other, to achieve disambiguation without reducing the coverage of the grammar. The importance of such principles must be assessed by the grammar writer along with the principles themselves; note that the relative importance of the agreement and ordering constraints mentioned above could vary from language to language.

A related problem is the representation of the very common case where a principle is considered universal but allows specific exceptions that are licensed by particular other words or constructions. In the face of such mutually contradictive rules, two alternative strategies are possible. The *exception handling* technique writes a rule that enforces the principle but also recognizes the exceptions. Typically this is done in the form of an implication that has additional cases disjoined to its conclusion. In contrast, the *shootout* technique simply writes both rules and gives them different scores.

Let us consider a simplified phenomenon from German. Assume that nouns (category NN) need determiners (label DET), but if the regent is a preposition that has a determiner fused into it (APPRART) then no determiner is required (or, indeed, allowed). The rule would explicitly make an exception for this case:

```
{X:SYN} : 'NN needs DET' : 0.5 :
  X↓cat = NN
  ->
  has(X↓id, DET) | X↑cat = APPRART;
```

The alternative is to cast both principles as individual rules, one of which is stronger than the other and will therefore ‘win’ if they are ever tested against each other. In the example case, one would simply require the determiner unconditionally, but forbid it absolutely if the APPRART is present:

```
{X:SYN} : 'NN needs DET' : 0.5 :
  X↓cat = NN -> has(X↓id, DET);

{X!SYN/Y!SYN} : 'APPRART conflicts with DET' : 0.0 :
  X↑cat = APPRART -> Y.label != DET;
```

This achieves much the same effect: normally nouns require their determiner and complain when it is absent, but if an APPRART is the regent, there can be no determiner, so the analysis without one is still optimal.

The difference between the two approaches lies in the response of the grammar when an exception actually occurs. With exception handling, the exception is silently accepted. In a shootout, the exception is accepted but makes ‘noise’, in the form of a failure of a weaker constraint. This may or may not be a disadvantage. For instance, when transformation-based parsing uses conflicts to decide whether to proceed and where, it will inevitably be misled by the inevitable conflict that is produced. Ideally it will quickly notice that the conflict cannot be solved and ignore it, but this does not always happen. In that case, unnecessary work is created that usually takes longer than it would have done to evaluate the longer exception-aware formula.

A related point is that the weaker constraint might also mislead a user of the parser. Since the more important rule — no DET if APPRART is present — is *not* violated, it will not show up in the conflict list, and therefore there is no clear indication of *why* the determiner rule failed. It might be either a language error or an exceptional situation, with no quick way to distinguish the two, since both cases are uniformly penalized by the same amount. This is not a problem if one is interested only in the accuracy of the language model in its entirety; however, if one wants to use the absolute score of individual analyses as a rough indicator of acceptability, merely unusual cases will be classified as errors, when they should really be considered correct.

6 A comprehensive grammar of German

We have developed a grammar for the WCDG formalism that is intended to cover the entire realm of modern German. As an example of our dependency representation, consider again the analysis of the sentence *"Man hat uns ein Angebot gemacht, das wir nicht ablehnen konnten." (They made us an offer we could not refuse.)* in Figure 1. Note that there are two instances of *nonprojective* structure in this sentence, i.e. the vertical projection lines must cross dependency edges in two places no matter how the tree is drawn. This is often considered a problem for parsers that operate on some variant of context-free derivation structures; in contrast, WCDG does not require its structures to be projective (although constraints to that effect can easily be written, and then have a considerable disambiguating effect). We can thus represent phenomena such as German auxiliary groups or extraposed relative sentences faithfully by writing constraints that enforce projectivity in general, but make exceptions for edges with these particular labels.

The referential relationship between the relative pronoun 'das' and its antecedent 'Angebot' cannot be represented on the syntax level, since both words already function as direct objects. Instead, it is represented by an additional edge which connects the two words on an extrasyntactical 'reference' level. The agreement of gender and number that must hold between the two words can thus be checked directly.[6]

The grammar currently consists of about 700 handwritten constraints, although much information is lexicalized, in particular valence information of verbs and prepositions. An extensive lexicon of full forms is used that includes all closed-class forms of German. Among the open-class forms, around 6,000 verb stems and 25,000 noun stems are covered; compound forms can be deduced from their base forms on the fly to deal with the various types of German compounds. As a last resort, lexicon templates provide skeleton entries for totally unknown words; these only contain the syntactic category and underspecified values for case, number etc. In the experiment reported here, 721 of 16649 tokens had to be

[6] Although other pronouns, and possibly nouns, can refer to other words, these relationships are usually outside the scope of a sentence-based parser because they transcend sentence boundaries. Therefore, this grammar describes reference edges *only* for the relative pronouns.

hypothesized from such templates, such as personal names and foreign-language material.[7]

So far we have considered phenomena from about 28,000 sentences from various text types, such as newspaper text, appointment scheduling dialogues, classical literature, law texts, and online newscasts. Although obscure extra-grammatical constructions are still occasionally encountered, the great majority of new input is covered accurately, and we do not expect major changes to the existing rules to become necessary. Many problem cases involve constructions that are probably impossible to represent accurately in a word-based dependency tree, such as complicated conjunctions or heavy ellipsis.

We estimate the overall effort for our grammar of German at about 5 work-years. A large part of this work, however, consisted in creating manual syntax annotations for testing the predictions of the constraint model; this corresponds to the effort needed to create the treebank that a stochastical parser is then trained on. Another very time-consuming task was to collect and classify open-class lexicon entries with respect e.g. to their inflection and valence features. While helpful for disambiguation in many cases, this information is not strictly necessary for running the parser.

7 Evaluation

Exact solution of the optimization problem posed by a large constraint grammar is often computationally infeasible for long input sentences, so that incomplete solution methods must usually be employed. Therefore, in addition to *model errors*, where the linguistic intuition contradicts the model defined by the actually implemented grammar, *search errors* can occur, where the parser output in turn differs from the modelled optimum because the heuristic optimization algorithm terminated in a local peak.

In the case of the earlier example sentence, no problems occur: the grammar assigns the highest score to the desired analysis shown in Figure 1, and the solution algorithm actually finds it. In general, this is not always the case, particularly for long sentences. Although all individual cases we have investigated suggest that the accuracy of the model established by our grammar is higher than the actual performance of the parser (i.e., search errors decrease rather than increase the overall accuracy), this is not of practical use because the better solutions cannot be efficiently computed. Therefore, all results reported in this paper use the strictest possible measure: only the accuracy of the *computed* structures with respect to the *intended* (annotated) ones is reported.

Since our parser always returns exactly one complete syntax structure, recall and precision are both identical to the percentage of correct versus incorrect dependency edges. To perform a more detailed evaluation we transformed 1000 phrase structure trees from the German NEGRA phrase treebank of German

[7] In theory, an unknown word could be of any open class, and thus introduce very great ambiguity; but most of these alternatives are usually discarded by the POS-tag preprocessing.

newspaper text to dependency structures automatically, with only few edges corrected to conform to our annotation guidelines [13]. The accuracy of parsing is then measured by counting the number of correctly computed dependency edges. Input is first annotated with part-of-speech tags by the statistical trigram tagger TnT, and some typical errors of the trigram model are automatically corrected by about 20 handwritten rules. The scores of the tagger are integrated into the constraint grammar with a constraint that disprefers word forms with improbable categories.

We employ a heuristic transformation-based solution method that first constructs a complete syntax tree in linear time, and then tries to transform it so as to remove errors that were diagnosed by violated constraints. Three transformation passes are made over each sentence; in the first phase, only subordinations between nearby words are allowed, and the resulting partial trees are recombined by their roots in the second phase. A third pass then tries to repair any remaining errors; this is the only phase that investigates the entire space of possible subordinations. We have found that the phase-based approach yields better results than tackling the full optimization problem to begin with [14].

Table 1 gives the results; altogether 89.0% of all dependency edges are attached correctly by the parser. This figure drops to 87.0% if we also demand the correct label at the dependency edges. Sentence length correlates with a slowly increasing number of parsing errors; to a certain degree this simply reflects the greater number of plausible but wrong subordinations in long sentences. Also, the incompleteness of the solution algorithm becomes more and more relevant: as the search space increases, more alternatives are overlooked.

In general, these results are near the performance of the Collins parser for English [4], but somewhat below the results of Tapanainen and Järvinen [15], who report precisions above 95% (also for English), but do not always attach all words. More recently, results for German have been published also. Dubey and Keller [16], analysed 968 sentences from the same section of the NEGRA corpus newspaper text as we used above, but restricted the test set to sentences with at most 40 words. Reimplementing Collins' parser for German and training it on the remaining sentences of the NEGRA treebank, they only achieved a labelled precision and recall of 66.0%/67.9% for dependency structures automatically derived from their phrase structure trees. Although the NEGRA treebank is clearly smaller if compared to the size of the Penn Treebank for English, this surprisingly large difference suggests that German syntax is considerably more difficult to analyse for this kind of parser than English. Dubey and Keller improved the parsing model to 70.9%/71.3% by considering sister-head dependencies instead of head-head dependencies.

Unfortunately, a direct comparison of these measures and our own results is not possible, since they refer to differently derived gold standard annotations. Moreover, one needs to keep in mind that

- the parser of Dubey and Keller was trained using only the restricted information which can be derived from a treebank automatically, while our WCDG parser is based an a handcrafted grammar and supported by a large lexical data base, and

– they compare gold standard annotations and parsing results which have both been derived automatically from phrase structure trees, while we rely on derived gold standard annotations but compare them with directly parsed dependency trees.

Nevertheless, Table 2 gives a rough indication of the level of quality that currently can be achieved by means of the two different approaches. Schielen [17] also trained a stochastic phrase structure parser on the NEGRA treebank, but provided it with a richer treebank annotation and an external lexicon as an additional information source. Thus, these results seem much better suited to be compared to ours. The additional information available to the parser also explains the remarkably better results than those obtained by Dubey and Keller. Nevertheless they are still worse than the ones achieved by means of our WCDG parser.[8]

test set	# of sent.	edges	lab. edges
all sentences	1000	89.0%	87.0%
$\leq$60 words	998	89.1%	87.1%
$\leq$40 words	963	89.7%	87.7%
$\leq$20 words	628	92.3%	90.1%
$\leq$10 words	300	93.4%	91.0%

Table 1. WCDG parsing results for NEGRA sentences.

		constituent structures			dependency structures		
	test set	precision	recall	f-score	precision	recall	f-score
Dubey & Keller [16]	$\leq$ 40 words	70.9%	71.3%	71.1%	—	—	76.1%
Foth et al. [1]	$\leq$ 40 words	—	—	—	87.7%	87.7%	87.7%
Schielen [17]	all sentences	—	—	69.4%	—	—	81.7%
Foth et al. [1]	all sentences	—	—	—	87.0%	87.0%	87.0%

Table 2. Comparison to other accuracy results on the NEGRA corpus. All numbers are given for labelled trees.

[8] Note that in this case the evaluation has been carried out on the full set of 1000 sentences.

8 Further experiments

A WCDG provides remarkable flexibility with respect to aspects of parsing behaviour. In particular, although the exact solution of the optimization problem that it poses would theoretically require exponential runtime, good results can also be achieved with much less resources. In fact, an arbitrary time limit can be set for the parsing process because the algorithm exhibits the *anytime property*: processing can be interrupted at any time if necessary, and the current analysis returned. This allows a trade-off to be made between parsing time and quality of results, since improvements to the score of an analysis generally coincide with a better accuracy.

For the results in Table 1, we allowed the transformation process to spend up to 600 seconds on each sentence (on 1533-MHz Athlon PC processors), and it actually terminated within less than 70 seconds on the average.[9] If the time limit is lowered, accuracy slowly decreases, as shown in Table 3.

time limit	avg. time used	edges	lab. edges
600 seconds	68.0s	89.0%	87.0%
400 seconds	59.3s	88.7%	86.8%
200 seconds	44.9s	88.2%	86.2%
100 seconds	31.8s	87.1%	85.0%
50 seconds	21.6s	84.6%	82.3%

Table 3. WCDG parsing results with reduced runtime limit.

We also investigated the generality of our model of German, originally developed to parse the text of online technical newscasts, by parsing a variety of other text types. Table 4 gives results of analysing various corpora under the same conditions as those in Table 1. In general, performance is measurably affected by text type as well as sentence length, but the accuracy is comparable to that on the NEGRA corpus. Classical literature was notably more difficult to parse, since it employs many constructions considered marked or extragrammatical by our current model of German. Transcriptions of spontaneous speech also pose some problems, since they contain many more sentence fragments than written text and no disambiguating punctuation.

A particular advantage of a language model composed of many cooperating constraints is that any single rule is not vital to the operation of the grammar. In fact, since constraints forbid rather than license particular structures, a missing rule can never cause parsing to fail completely; at most it can increase the search space for a given sentence. This means that a grammar can be applied and tested

[9] Establishing the constraint satisfaction problem in the first place, i.e. precomputing all possible dependency edges, takes additional time, but less than 10% of the parsing time on average.

text type	sentences	avg. length	edges	lab. edges
trivial literature	9547	14 words	93.1%	91.1%
law text	1145	19 words	88.8%	86.7%
verbmobil dialogues	1316	8 words	90.3%	86.3%
online news	1894	23 words	89.8%	88.1%
serious literature	68	34 words	78.0%	75.4%

Table 4. WCDG parsing results on different text types.

even while it is still under construction, so that a realistic grammar for an entire language can be developed step by step.

To demonstrate this robustness against omitted rules, we deliberately disabled entire groups of constraints of the grammar, one group at a time, and repeated the comparison experiment of Section 7. Table 5 shows the results. For each group of constraints the general purpose is given as well as the informal version of an example constraint. The importance of each constraint group is given as the ratio between the structural correctness achieved with and without the constraints from this group.

class	purpose	example	importance
init	hard constraints	appositions are nominals	3.70
pos	POS tagger integration	prefer the predicted category	1.77
root	NIL subordinations	only verbs should be tree roots	1.72
cat	category cooccurrence	prepositions do not modify each other	1.13
order	word-order	determiners precede their regents	1.11
proj	projectivity	disprefer nonprojective coordinations	1.09
exist	valency	finite verbs must have subjects	1.04
punc	punctuation	subclauses are marked with commas	1.03
agree	rection and agreement	subjects have nominative case	1.02
lexical	word-specific rules	“entweder” requires following “oder”	1.02
dist	locality principles	prefer the shorter of two attachments	1.01
pref	default assumptions	assume nominative case by default	1.00
sort	sortal restrictions	“sein” takes only local predicatives	1.00
uniq	label cooccurrence	there can be only one determiner	1.00
zone	crossing of marker words	conjunction must be leftmost dependent	1.00

Table 5. Measuring the effect of different constraint classes.

As expected, the most important type of constraints (in that their omission degrades performance the most) is the “init” group, into which most hard constraints fall. Parsing here suffers from a vastly increased initial ambiguity that often prevents the parser from finding the correct solution even if it is still theoretically optimal. POS preprocessing turns out to be very important as well, again because it quickly closes huge but implausible search spaces. Rules that disprefer multiple NIL subordinations effectively prevent the parser from taking the 'easy' solution and treating words that it cannot attach as additional roots.

Each other group of constraints has comparatively little effect on its own. This is often because the constructions that a constraint forbids are not possible when parsing a particular corpus. For instance, the group "uniq" contains a constraint that forbids noun phrases with more than one determiner, but if the test set does not contain any noun phrases with more than one article, such analyses are never possible in the first place. The rule is therefore largely ineffective here, although it is of course rather important in the general case.

9 Related Work

Except for the approaches of Dubey and Keller [16] and Schielen [17] there seem to be no other attempts to evaluate a syntactic parser of German on a gold standard annotation using the common PARSEVAL methodology. Accordingly, there is no standard setting available so far which could facilitate a direct parser comparison, similar to the established Penn Treebank scenario for English. However, there have been evaluation efforts for more shallow types of syntactic information, e.g. chunks or topological fields.

Parsing into topological fields like Vorfeld, Mittelfeld, and Nachfeld usually is considered a somewhat easier task than the construction of a full-fledged constituency analysis, because it clearly profits from the easily identifiable position of the finite verb in a German sentence and avoids any decision about the assignment of syntactic functions to the constituents. Braun [18] presents a rule-based approach to topological parsing of German and reports a coverage of 93.0%, an ambiguity rate of 1.08, and a precision/recall of 86.7% and 87.3% respectively on a test set of 400 sentences. Becker and Frank [3] trained a stochastic topological parser on structures extracted from the NEGRA-Treebank and tested it on 1058 randomly selected sentences with a maximum length of 40 words. They achieved a labelled precision/recall of 93.4% and 92.1%.

Chunking, i.e. the segmentation of a sentence into pieces of particular type (usually NPs and PPs), is an approach to shallow sentence processing sufficient for many practical purposes. Brants [19] used cascaded Hidden Markov Models to chunk sentences into a hierarchy of structurally embedded NPs and PPs with a maximum tree depth of nine. He achieved a precision/recall of 91.4% and 84.8% in a cross evaluation on the 17,000 sentences of the NEGRA-Treebank. Schmid and Schulte im Walde [2] trained a probabilistic context-free grammar on unlabelled data and used it to segment sentences into NP chunks. They evaluated the system on 378 sentences from newspaper text and obtained a precision/recall of 93% and 92% if only the range of a chunk is considered, which decreased to 84% and 83% if also the case of an NP has to be determined.

Special evaluation methodologies have been designed for several parsers of German. Among them is the one used to evaluate the quality improvement during the development of the Gramotron parser [20]. The method is partly motivated by lexical semantic clustering, the ultimate goal of the project, although the authors admit that a direct evaluation of head-to-head dependencies as proposed by Lin [21] would be more appropriate. Besides avoiding overtraining by monitoring the cross-entropy on held-out data, a linguistic evaluation was carried out focussing on selected clause types. It is based on a randomly sampled subcorpus of 500 clauses and measured the quality of determining noun chunks and subcategorization types. The precision of chunking reached 98% if only the range was considered and decreased to 92% if the chunk type was also included. The subcategorization type of verbs was computed with a precision between 63% and 73% depending on the kind of clauses under consideration.

Langer [22] evaluated a GPSG parser with the capability of additional processing rules (of unlimited generative power) on 62,073 sentences from German newspaper text. Of the sentences with 60 words or less, 34.5% could be analysed, with an average of 78 solutions per sentence. On a self-selected test corpus with an average length of 6 words, coverage rose to 83%. Only an indirect measure of accuracy is given: in 80.8% of the newspaper sentences, giving the parser access to partially annotated target structures (chunks and some POS information) did not change its output.

Neumann and Flickinger [23] evaluated their HPSG-parser based on the DOP-model with 1000 sentences from the Verbmobil-domain. They measured coverage (70.4% if combined), and the runtime performance of the system.

Wauschkuhn [24] developed a two step parser computing a clause level structure which clause-internally is kept completely flat. The system is evaluated on 1097 newspaper sentences only with respect to its coverage (86.7% if run on manually tagged input and 65.4% for automatically tagged text) and the achieved degree of disambiguation (76.4% for manually and 57.1% for automatically tagged input).

Among the different constraint-based approaches to natural language parsing, Property Grammar (PG) [25] comes closest to the idea of CDG. Here, constraints are defined for constructions, which are derived from a certain pattern of satisfied or violated constraints for subsets of word forms from the input sentence. Although PG does not require to first build a structural description before the constraints can be evaluated (instead the structure is constructed from the constraint evaluation results) the parser nevertheless first needs to select appropriate subsets of word forms as a basis for constraint application. Implicit constraints on this selection mechanism (e.g. adjacency) crucially determine the complexity of this matching procedure. In contrast, constraints in WCDG can be directly evaluated on at most n^2 dependency candidates where n is the length of the sentence under consideration.

Given the constraint evaluation results, the PG parser selects those constructions which caused the least number of constraint violations. As in WCDG, this notion of defeasible constraints introduces a certain degree of robustness, although the binary weighting scheme of PG does not provide for a fine grained arbitration between differently important constraints which, however, is possible with the scoring mechanism of WCDG.

10 Conclusions

A competitive parsing system for unrestricted German input has been described that is largely independent of domain and text type. Total coverage is ensured by means of defeasible, graded constraints rather than hard grammar rules. The method of using individual constraints for different grammatical principles turns out to be useful for ongoing grammar development, as the parser can be run normally from the beginning and aspects of the working language can be modelled one at a time.

We find that using a powerful formalism that allows all conceivable rules to be actually used allows for better parsing than using a tractable but less expressive one: even a theoretically intractable formalism is useful in practice if its model can be approximated well enough. While casting syntax analysis as a multidimensional optimization problem leads to high requirements of processing time, this inefficiency is somewhat mitigated by the anytime property of the transformational optimizer that allows the user to limit the desired processing time.

A greater concern is certainly the great amount of expert knowledge needed to create a complete constraint grammar. We propose that for deep analysis of German syntax, the greater precision achievable through handwritten rules justifies the extra effort compared to approaches based purely on machine learning. Although automatic extraction of constraints from annotated corpora remains a goal of further research, so far we have achieved greater success by hand-selecting and scoring all grammar rules, since this allows the grammar writer to make conscious decisions about the relative importance of different principles.

An online demonstration of our system can be accessed at `http://nats-www.informatik.uni-hamburg.de/Papa/ParserDemo`.

The source code of the parser and the grammar is available under the General Public License at `http://nats-www.informatik.uni-hamburg.de/download`.

Acknowledgements

This research was partially supported by Deutsche Forschungsgemeinschaft under grant Me 1472/4.

References

1. Foth, K., Daum, M., Menzel, W.: A broad coverage parser for german based on defeasible constraints. In: Proc. 7. Konferenz zur Verarbeitung natürlicher Sprache, KONVENS-2004, Vienna, Austria (2004) 45–52
2. Schmid, H., Schulte im Walde, S.: Robust German noun chunking with a probabilistic context-free grammar. In: Proc. 18th Int. Conf. on Computational Linguistics, Coling-2000, Saarbrücken, Germany (2000) 726–732

3. Becker, M., Frank, A.: A stochastic topological parser for German. In: Proc. 19th Int. Conf. on Computational Linguistics, Coling-2002, Taipeh, Taiwan (2002) 71–77
4. Collins, M.: Head-Driven Statistical Models for Natural Language Parsing. PhD thesis, University of Pennsylvania, Philadephia, PA (1999)
5. Schröder, I.: Natural Language Parsing with Graded Constraints. PhD thesis, Hamburg University, Hamburg, Germany (2002)
6. Maruyama, H.: Constraint dependency grammar. Technical Report RT0044, IBM Research, Tokyo Research Laboratory (1990)
7. Tsang, E.: Foundations of Constraint Satisfaction. Academic Press, London (1993)
8. Pollard, C., Sag, I.A.: Head-Driven Phrase Structure Grammar. The University of Chicago Press, Chicago (1994)
9. Daum, M., Menzel, W.: Parsing natural language using guided local search. In: Proc. 15th European Conference on Artificial Intelligence, ECAI-2002, Lyon, France (2002) 435–439
10. Foth, K., Menzel, W.: A transformation-based parsing technique with anytime properties. In: Proc. International Workshop on Parsing Technologies (IWPT-2000), Trento, Italy (2000) 89–100
11. Foth, K., Menzel, W., Schröder, I.: Robust parsing with weighted constraints. Natural Language Engineering (forthcoming)
12. Foth, K.A.: Writing weighted constraints for large dependency grammars. In: Proc. Recent Advances in Dependency Grammars, COLING-Workshop 2004, Geneva, Switzerland (2004)
13. Daum, M., Foth, K., Menzel, W.: Automatic transformation of phrase treebanks to dependency trees. In: Proc. 4th Int. Conf. on Language Resources and Evaluation, LREC-2004, Lisbon, Portugal (2004) 1149 – 1152
14. Foth, K., Menzel, W.: Subtree parsing to speed up deep analysis. In: Proc. 8th Int. Workshop on Parsing Techniques, IWPT-2003, Nancy, France (2003) 91–102
15. Tapanainen, P., Järvinen, T.: A non-projective dependency parser. In: Proc 5th Conference on Applied Natural Language Processing, ANLP-1995, Washington, D.C. (1997) 64–71
16. Dubey, A., Keller, F.: Probabilistic parsing for German using sister-head dependencies. In: Proc. 41st Annual Meeting of the Association of Computational Linguistics, ACL-2003, Sapporo, Japan (2003) 96–103
17. Schielen, M.: Annotation strategies for probabilistic parsing in german. In: Proc. 20th Int. Conf. on Computational Linguistics, Coling-2004, Geneva, Switzerland (2004) 390–396
18. Braun, C.: Parsing German text for syntactico-semantic structures. In: Prospects and Advances in the Syntax/Semantics Interface, Proc. of the Lorraine-Saarland Workshop, Nancy (2003)
19. Brants, T.: Cascaded markov models. In: Proc. 9th Conf. of the European Chapter of the ACL, EACL-1999, Bergen, Norway (1999) 118–125
20. Beil, F., Prescher, D., Schmid, H., Schulte im Walde, S.: Evaluation of the Gramotron parser for German. In: Proceedings of the LREC Workshop: Beyond PARSEVAL, Las Palmas, Gran Canaria (2002) 52–59
21. Lin, D.: A dependency-based method for evaluating broad-coverage parsers. In: 14th Int. Conf. on Artificial Intelligence, IJCAI-1995, Montréal, Québec, Canada (1995) 1420–1427
22. Langer, H.: Parsing-Experimente: praxisorientierte Untersuchungen zur automatischen Analyse des Deutschen. Peter Lang, Frankfurt am Main (2001)

23. Neumann, G., Flickinger, D.: HPSG-DOP: Data-oriented parsing with HPSG. In: Unpublished manuscript, presented at the 9th Int. Conf. on HPSG, HPSG-2002, Seoul, South Korea (2002)
24. Wauschkuhn, O.: The influence of tagging on the results of partial parsing in German corpora. In: Proc. 4th Int. Workshop on Parsing Technologies, IWPT-1995, Prague/Karlovy Vary, Czech Republic (1995) 260–270
25. Blache, P., Balfourier, J.M.: Property grammars: a flexible constraint-based approach to parsing. In: Proc. Int. Workshop on Parsing Technology, IWPT-2001, Beijing, China (2001)