Natural Language Engineering **1** (1): 1 – 25. Printed in the United Kingdom

Robust Parsing with Weighted Constraints

Kilian Foth, Wolfgang Menzel, Ingo Schröder

(*Received 1 April 2003*)

Abstract

Based on constraint optimization techniques an architecture for robust parsing of natural language utterances has been developed. The resulting system is able to combine possibly contradicting evidence from a variety of information sources, using a plausibility-based arbitration procedure to derive fairly rich structural representations, comprising aspects of syntax, semantics and other description levels of language. Results of a series of experiments are reported which demonstrate the high potential for robust behaviour with respect to ungrammaticality, incomplete utterances, and temporal pressure.

1 Introduction

Robustness is a fairly general aspect of system behavior which imposes a kind of stability requirement on the range of acceptable responses: A robust system should exhibit an expectation-preserving behavior even under adverse environmental conditions. This notion is equally well applicable to a robust state of health, a robust car or a robust natural language processing system. For information processing systems, which in general establish a functional relationship between the input of a system and its output, robustness can be described as the requirement that even relatively large distortions at the input of a system should result in a comparatively small deviation from the expected behavior at the output. Thus, talking qualitatively and quantitatively about robustness requires at least four different things:

- a specification of the expected (or normal) behavior,
- a specification of possible adverse factors,
- a metric to describe the degree of input distortion, and
- a metric to describe the deviation of observed results from the expected ones.

Whereas for most natural language applications an at least intuitive agreement can be expected with respect to the first and fourth precondition, the second one needs a more careful consideration. It refers to the multidimensional nature of robust behavior in almost any practical setting which makes it necessary to spell out precisely, against which influences robust behavior is expected. Having a closer look at the human language processing system we can identify a fairly broad range of dimensions which contribute to its apparent robust behaviour. Among them are robustness

- against noise in the sensory channels, caused by e. g. blurred writing, crosstalk, etc.,
- against the inherent uncertainty of the stimulus, typically found in speech and hand written text,
- against variance between speakers/writers,
- against insufficient competence,
- against ill-formed or incomplete input as defined by some normative standard, and
- against (external or internal) temporal pressure.

Not all these phenomena have attracted the same degree of attention when it comes to the design and implementation of technical solutions for natural language processing. The problem of dealing with *insufficient competence* or extragrammaticality, for instance, has mainly been considered from the somewhat biased perspective of maximizing the overall accuracy of a processing component on a given corpus. Here, machine learning approaches, which extract the necessary model information directly from a corpus of annotated utterances, have proven quite successful, c.f. (Collins 1999), (Charniak 2000). As long as the necessary training data are available, they offer an attractive solution to the knowledge acquisition bottleneck traditional approaches suffer from. On the other hand, current machine learning methodology adopts a fairly static perspective on the problem of knowledge acquisition, by considering closed collections of data and strictly separating training from testing. Both assumptions seem rather unnatural if compared to the way humans learn and use their language(s).

Closely related to the issue of knowledge acquisition is the development of mechanisms that would allow a language processing system to actively deal with an insufficient coverage of lexis, syntax or semantics. Again, the human model shows a remarkable ability to cope with such a situation, be it as a (foreign) language learner or a novice to a specific language for a particular domain. Under such conditions a partial analysis might well be sufficient for most information seeking purposes to successfully bootstrap an approximate understanding while processing the same input repeatedly.

This scenario shares some similarities with the task of searching a particular piece of information in a large body of text: While in the first case the processing is constrained by the impoverished competence of the language processor itself, now the depth of analysis can be deliberately reduced to shallow structures, whenever the relevance of the data under consideration is expected to be low.

In contrast to the astonishing flexibility which the human language processor shows under such conditions, computational approaches to shallow analysis (e. g., tagging (Church 1988),(Charniak et al. 1996) or chunking (Abney 1991)), are geared towards a predefined level of representational granularity and can hardly be adapted to actual information needs or available computational resources in a dynamically changing environment.

Given the obvious success of corpus-based training methods when applied to the processing of large amounts of unrestricted text it seems tempting to apply similar techniques to the problem of *ungrammatical input* as well. If only a large enough collection of ill-formed utterances could be provided, the system should be able to acquire recurrent patterns of language use from the available data. However, this approach of training the distortion into the model remains unsatisfactory for a number of reasons:

- Compiling a representative sample of distorted data happens to be far more difficult compared to the collection of close-to-standard utterances, since ill-formed language is highly specific to the circumstances under which it has been produced: Speech recognition results differ from non-native data, which in turn are clearly influenced by the first language of the speaker/writer and her level of language command.
- The idiosyncratic nature of ill-formed language data will inevitably lead the system to adapt to particular forms of deviation while neglecting other ones.
- Without special provision the system is likely to loose the ability to distinguish the norm from the deviation, while adapting to the most frequently observed examples.

Alternatively, one might want to have a system which maintains the (possibly fuzzy) notion of a standard language despite being exposed to frequent deviations, thus being able to deal with the tension between norm and observation in order to detect, diagnose, and correct inappropriate constructions.

Finally, a graceful response to *temporal pressure* is another important robustness feature of the human language processor, which allows it to dynamically adapt to external or internal speed requirements like varying speech rates or skimming through. In such cases the language processor should be able to trade certain quality features of the expected processing results (e.g. correctness, depth or completeness of understanding) against processing speed.

To achieve a comparable temporal behavior in a computational approach, anytime algorithms (Boddy and Dean 1994) need to be devised. They are characterized by a performance profile which predicts the expected quality of processing results as a monotonically growing function of time. Processing can be terminated whenever the expected increase in quality no longer can be justified by the usually decreasing utility of results. By combining the performance profile with external utility requirements an optimal upper threshold of processing time can be determined prior to the computation itself. Since the system is granted a fixed amount of time and a result with a quality level as specified by the performance profile is given in return, the corresponding class of algorithms is called contract algorithms.

An even stronger anytime property requires the algorithm to provide a result of the predicted quality level at an *arbitrary* point in time. If available, such an interruptable algorithm can be integrated in a dynamic scheme of arbitration, where a kind of meta control component constantly monitors the progress of computation and decides whether to stop the computation or to continue for yet another period. In more advanced approaches meta control can even modify parameters of the analysis procedure with the goal of enforcing a speed up. To make sure that the system does not collapse under such temporal pressure, the underlying processing mechanism, however, must provide a considerable degree of robustness.

If robustness turns out to be such a multidimensional phenomenon, the question arises whether we need completely different mechanisms to deal with all those different aspects of system behaviour. It is the underlying assumption of the research described in this paper that this is not necessarily so.

To back up this claim, an architecture for parsing natural language utterances is presented which is able to address several aspects of robust processing by means of a single computational framework. Using weighted constraints to determine the most plausible structural description, a solution has been achieved which combines robust and error-sensitive treatment of ill-formed and incomplete utterances, as well as robustness against external temporal pressure, while maintaining the ability to detect and diagnose deviations from the norm. It is this unique combination of features, not the features considered in isolation, which makes the approach a promising starting point for the development of a new generation of natural language systems intended to more closely resembling the language processing capabilities of the human model.

Section 2 discusses essential prerequisites of a robust processing architecture, whereas Section 3 presents their realization by means of a constraint optimization approach. Sections 4 and 5 describe the results of a series of initial studies, carried out to explore the system's potential for robust processing of natural language.

2 Prerequisites for robust behavior

Three prerequisites seem to be crucial for achieving a robust solution for typical symbolic information processing tasks in general, and natural language processing in particular:

- the ability to deal with inconsistent axiom systems,
- the ability to carry out preferential reasoning, and
- the availability of structural redundancy.

We will discuss these issues in the following sections in more detail.

2.1 Accommodating inconsistency

Inconsistency is pervasive in natural language. It manifests itself overtly as conflicts between observable linguistic data on the one hand and regularities of the language system on the other. This is most obvious in case of genuine grammatical errors (violations of syntactic conditions):

**Why decided he to leave?*

but also in marked usages as topicalization phenomena (violation of word order regularities):

Nice things you have been telling me!

or metaphorical use (violation of selectional restrictions):

This house eats up all my money.

Inconsistencies can also be found within the language system itself. Here, they mostly arise from the existence of clearly contradictory preferences, as in the case of

- the resolution of anaphorical references which tends to prefer a short distance to the antecedent

  My brother hates his ***boss***$_i$ *because* ***he***$_i$ *is unable to hide his feelings.*

  but can be guided by syntactic and semantic parallelisms as well

  My ***brother***$_i$ *hates his boss and* ***he***$_i$ *is unable to hide his feelings.*

- German relative clauses extraposed from a discontinuous verb, where the tendency to keep the referential distance short:

Das	***hängt***	*von*	*einem*	*Termin*$_i$,
This	depends$_{\text{root}}$	on	an	appointment

den$_i$	*ich*	*erst*	*morgen*	*erfahre,*	***ab***.
which	I	not until	tomorrow	learn,	depends$_{\text{prefix}}$

  This depends on an appointment, which I will not learn until tomorrow.

  is in strong conflict with the tendency to keep the verb group together:

  Das ***hängt*** *von einem Termin*$_i$ ***ab****, den*$_i$ *ich erst morgen erfahre.*

  The desire to keep the two components of the verb as close as possible in oral communication even leads to

  $^?$*Das* ***hängt ab*** *von einem Termin*$_i$*, den*$_i$ *ich erst morgen erfahre.*

  which satisfies both minimal distance requirements, but causes a severe violation of the canonical word order in German main clauses.

- the dependence of ordering regularities in the German *mittelfeld* on the syntactic material being pronominalized or not, which can be described efficiently using the following possibly conflicting requirements:

  the subject should precede the object
  only material in topic position should be pronominalized
  a pronominalized subject must precede a pronominalized object

- a mismatch between grammatical and natural gender, where a reference according to natural gender

  Das ***Mädchen***$_{\text{neuter}}$ *spielt mit* ***ihrer***$_{\text{feminine}}$ *Puppe.*
  The girl plays with her doll.

  is even preferred over the grammatically correct gender agreement

  Das ***Mädchen***$_{\text{neuter}}$ *spielt mit* ***seiner***$_{\text{neuter}}$ *Puppe.*

- all kinds of elliptical utterances where the general preference for complete utterances is in conflict with the Gricean maxim to be concise.

Inconsistencies are by no means an undesirable phenomenon of natural language. First of all, they contribute considerably to the efficiency and ease of language communication. In many cases inconsistencies even carry a communicative function, e. g., to attract the attention of the listener/reader by means of techniques like topicalization, contrastive stress, metaphor, or other expectation violations. Similarly, an incomplete utterance can signal whether the speaker wishes to hand over the initiative to the dialogue partner.

Being both frequent in and constitutive of natural language communication, inconsistencies cannot easily be worked around in a natural language processing system. In the ideal case, they should be treated explicitly, by making the system aware of conflicting requirements and able to arbitrate between them. Additionally, such a mechanism provides a considerable contribution to increase the robustness of the system, since conflicts between real input data and expectations about its ideal form as modelled in the grammar can be resolved without any further provision.

2.2 Preferential reasoning

Inconsistency can either be treated by (dynamically) extending the grammar beyond the "normally" required coverage (overgenerating) or by selectively neutralizing or retracting constraints. Both alternatives increase the space of structural interpretations for a given utterance considerably and require a suitable compensation. Usually this compensation is achieved by means of a plausibility measure, e. g. probabilities, activation degrees, belief or (un-)certainty factors. Such a numerical measure serves three different purposes: It defines an optimal interpretation of the input, it can be used to guide the search for the optimal solution and during searching it facilitates an arbitration between different sources of evidence. Now, inherently uncertain information can be integrated into the model which would not be available on a purely consistency-based account. In particular, a wide range of relatively weak contributions (ordering preferences, default cases etc.) are made available to guide the decision procedure.

Indeed, a closer inspection reveals that really strict rules are surprisingly rare in natural language. Fairly many regularities come with exceptions and actually turn out to be more of a preferential than a crisp nature. All the examples for language-internal inconsistencies mentioned in the previous section, actually can and need to be expressed as preferences. Other examples are defaults which in the absence of decisive evidence prefer one interpretation over the other. In German the subject tends to precede the direct object, but this is not at all a strict rule. It is used to obtain an interpretation, if a truly ambiguous sentence like

Die Frau$_{\text{subject}}$	sieht	die Tochter$_{\text{object}}$.
fem. sg.		fem. sg.
{ nom, acc }		{ nom, acc }

The woman sees the daughter.

is considered in isolation. As soon as stronger evidence like case markers, emphatic stress or contextual information is available, it can easily be overridden:

Die Frau$_{\text{object}}$	sieht	der Mann$_{\text{subject}}$.
fem. sg.		masc. sg.
{ nom, acc }		{ nom }

The man sees the woman.

Preferential reasoning must be considered a crucial contribution to the success of probabilistic models in language processing. By conditioning e. g. structural alternatives on different kinds of predictors like the head word, the structural context or the distance to the head word, a massively overgenerating model is introduced and constrained again according to the frequencies of structural patterns observed in the training data. Thus, the most frequent case is established as a kind of default which, however, can be overridden by less frequent observations, if they contribute to a more probable interpretation for the whole utterance. The same holds even for unseen cases, which receive a non-zero probability by means of smoothing and back-off techniques.

Preferential reasoning has also been adopted in Optimality Theory (Prince and Smolensky 1993) where principles of universal grammar are arranged in a strict hierarchy and a solution is searched for which violates the least important principle. However, defeasible principles are introduced conservatively and preferential knowledge is not used on a massive scale. Nevertheless, compared to purely generative models even this cautious use of preferences contributes to a remarkable increase in robustness: In general, a sentence is not rejected due to incompatibilities with the grammar, but gets assigned the best possible structural description.

2.3 Redundancy

Redundancy in general is known to be a major source of robustness in information processing systems. By utilizing redundant codes, channel noise can be tolerated or corrected (Cover and Thomas 1991). On the other hand structural redundancy might improve system dependability. Both mechanisms can be observed in natural language as well, where the redundancy of the code and the potential for structural redundancy among processing components are closely tied together.

This correspondence appears most obviously in the interplay between auditory and visual perception during speech comprehension. Evidence from the two independent channels is integrated seamlessly while maintaining their relative autonomy: Both modes, i. e., listening without seeing (e. g., communication over the telephone) and seeing without hearing (lipreading), are available in isolation. Nevertheless, highest robustness is achieved if they work together in a cooperative manner (Massaro 1998).

A similar phenomenon of synergy can be observed between syntax and semantics. Again, the two modes are (at least in extreme cases) available in isolation

- syntactic bootstrapping: deriving a rudimentary semantics from semantically void utterances based on strong syntactic evidence

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
Lewis Carroll, Jabberwocky

- semantic bootstrapping: deriving a syntactic structure from syntactically distorted utterances based on strong semantic evidence

  **We tonight cinema going.*

Under normal circumstances, however, syntactic and semantic evidence work closely together during human sentence comprehension and thus achieve an optimal performance due to their synergy (see e. g. (Crain and Steedman 1985) for the influence of contextual information and (Trueswell, Tanenhaus, and Garnsey 1994) for impact of thematic fit). In order to benefit most from such a mode of cooperation, it is important to avoid strong sequential dependencies which prevent the earlier processing steps from using information only available later on. As far as possible, all the components involved should work in parallel, to provide their evidence in a timely manner.

Whereas overgeneralization and preferential reasoning are fundamental for numerous approaches to robust parsing, the idea of using structural redundancy as an additional source of robustness has not attracted the same attention so far. To design an architecture which facilitates the necessary bidirectional propagation of evidence and the mutual compensation of partial deficits without making one component dependent on the contribution of the other, a weak coupling between them needs to be established. Unlike the crisp syntax-semantic interface in constraint-based grammars like HPSG (Pollard and Sag 1994)[1], where syntactic and semantic evidence is coreferenced in complex feature structures, a plausibility-based mapping between different representational levels of language would allow an interpretation even in cases of mismatch between syntactic and semantic evidence.

As became increasingly clear in the previous discussion, the three fundamental contributions to robust behavior, namely the accommodation of inconsistency, plausibility-based reasoning and the availability of structural redundancy, in fact are closely related to each other. Similar to the treatment of inconsistency, the introduction of structural redundancy also requires an arbitration mechanism based on plausibility scores. Thus, a kind of weighted combination of very different knowledge components needs to be established as a general principle throughout the system.

To meet this requirement, an integrated architecture for parsing natural language utterances has been developed which uses defeasible constraints on structural descriptions as the only means to express the necessary linguistic and extralinguistic knowledge. This architecture is described in the next section.

[1] Note that the notion of a constraint in HPSG differs from the notion of constraints as introduced in section 3.1. Whereas the first is defined over complex, recursive feature structures, the latter checks for admissible value assignments to a set of variables.

3 Parsing as a constraint optimization problem

3.1 Modelling language structure as dependencies

Weighted Constraint Dependency Grammars (WCDG) aim at finding the underlying relations between the words in an input utterance (Heinecke et al.1998). The principal idea of dependency grammar (Tesniere 1959) is that each word in the surface structure of an utterance *modifies* or *depends on* exactly one other word. This other word is called the *governor*.

As a rough approximation, one word is the governor of another word when it cannot be omitted without also omitting the other word. Thus, in the phrase “the cars” the word “cars” should be the governor, since it can appear without its determiner but not vice versa. The dependency relations in an entire utterance should form a complete tree, usually with the finite verb of the main clause as the root.

Since no tree-internal intermediary phrases are explicitly established, all syntactic structure is encoded as direct relations between words. This means that a few distinctions between different possible structures for an utterance cannot be easily expressed, in particular in complex coordinations with argument clusters. However, phenomena such as the ambiguous attachment of prepositional phrases can be handled very well.

In WCDG, this principle of structure as directed relations between words is extended to other levels of linguistic description. Quite different types of relations can be modelled, ranging from syntactic dependencies via anaphoric relationships to thematic roles and – in principle – even simple domain knowledge. However, the word becomes increasingly inappropriate as a unit of modelling the more abstract the level of analysis is, and therefore the emphasis of a WCDG is usually on describing syntactic structure.

Figure 1 shows an example analysis for an utterance where three different types of information are distinguished: syntax, semantics and reference. On the syntactic level, the relational analysis corresponds to a complete dependency analysis where each word (except the root) is subordinated under another word and the relations are annotated with labels (*subj, obj, det, ...*) which further characterize the relation.

For other levels of analysis, the relations do not constitute a coherent tree. Instead they resemble slot-filling constructions where only some words relate to other words. Thus on the semantic level, the verbs select words that can fill their thematic roles, but it is not possible to merge the relations in one tree-like description since some words, e. g., “he” in the example, actually relate to more than one governor. Finally, on the reference level of the example only the pronouns establish relations to other words. (Although nouns can also be viewed as referring to other words, the example does not establish a relation e.g. for “man” because the antecedent cannot be found in the sentence itself.)

Formally, all these relations between words on different levels are triples of the form (word,label,word). For each word and each level of analysis, exactly one of these triples is postulated. This means that relations like the thematic roles shown in the example cannot always be represented on one level of analysis; instead the different thematic roles possible should be modelled as different levels of analysis.

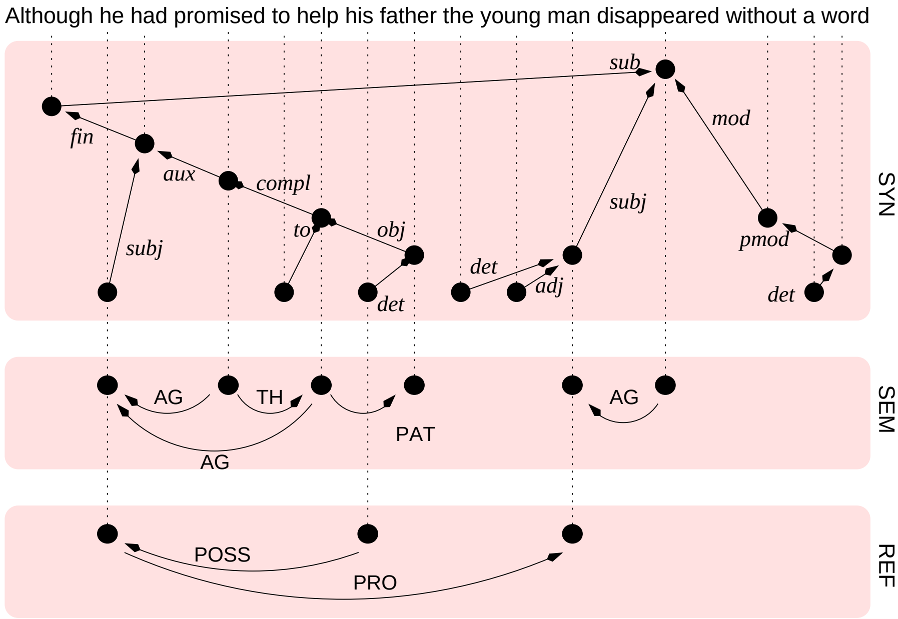

Fig. 1. Example of a multi-level dependency analysis

Conversely, when a word is not related to another word on one level (such as the semantically empty words on the semantic levels or the finite verb on the syntactic level), the special triple (word,label,NIL) is selected to represent this fact. This non-subordination is a perfectly valid choice, no different from an actual relation to another word, but NIL relations are usually omitted from diagrams to emphasize the actual structure of an analysis.

To establish such a triple on a particular level, a governor must be chosen as well as a label from the set of labels defined for this level. It is also necessary to select the particular lexical item that is to be assumed at this position. For example, Figure 1 implies that “help” is a verb, but “man” is a noun, both of which can be wrong in other contexts. In morphologically richer languages it can also be necessary to decide which of several homonymic variants of the same word is present, e.g. to distinguish a nominative from an accusative reading. Finally, when analysing spoken language, even the surface form of a word may be uncertain so that even more lexical ambiguity arises. WCDG is capable of operating on graphs of competing word hypotheses; the only requirement on a solution is that the set of lexical items ultimately assumed constitutes a single complete path through the entire length of the utterance.

Since all levels of processing use the same representational and computational framework, analysis need not be sequential like in more traditional architectures. Instead all levels can be worked on at the same time.

3.2 Writing linguistic rules as constraints

Most methods for automatic language analysis proceed according to rules in a formal derivation system that *allow* particular partial trees. Whether they proceed bottom-up or top-down, analysis is always driven by generative rules such as $S \rightarrow NP\ VP$. In contrast, WCDG does not have a concept of legal and illegal derivations; *all* triples (subordinations) of words in a sentence and labels defined by the grammar writer are initially possible. Rules are then written to *disallow* particular constructions; hence the entire process is one of elimination rather than generation.[2]

All rules in a WCDG are notated in the form of *constraints*, i.e. logical formulas which all subtrees must satisfy. The simplest constraints are sensitive to word categories and labels only; a common rule states that an object subordination can only subordinate a noun under a verb:

```
{X:SYN} : obj_category :
  X.label = obj  ->  X@category = N & X^category = V;
```

Here, the rule `obj_category` constrains all subordinations `X` with the label `obj`: The modifier of such a relation, notated as `X@`, must have the lexical feature `category` with a value of `N`, and the governor (`X^`) must carry the value `V`. If a relation does not satisfy this condition, it is disallowed.

Such conditions somewhat resemble generative derivation rules; note, however, that this constraint does not proscribe an order between the constituents. If desired, another constraint can be written to force the object to appear after the verb:

```
{X:SYN} : obj_position :
  X.label = obj  ->  distance(X@id, X^id) < 0;
```

Here the expression `distance(X@id, X^id)` returns the distance between the modifier and the governor of the relation `X`, measured from right to left.

Some linguistic rules are inherently concerned with more than two words; for example, where both a direct and an indirect object appear for the same verb, it can be postulated that the indirect object must come first:

```
{X:SYN, Y:SYN} : obj_order :
  X.label = obj & Y.label = iobj & X^id = Y^id
  ->
  distance(X@id,Y@id) > 0;
```

All of these constraints are enforced on each dependency edge or pair of edges in the entire dependency structure. For efficiency reasons, no more than two dependency relations can be constrained in the same rule. (Schröder et al. 2000) contains a fuller account of how to model different linguistic phenomena with this kind of universal constraints.

It is often inappropriate to model linguistic rules as strict constraints. For instance, the positional constraint shown above does not apply to German: an object

[2] This means that a WCDG cannot be used for language generation without an additional generative component.

can legitimately precede its verb, although more often it comes after the verb. For this reason WCDG allows constraints to bear a *score* between 0 and 1 that indicates the importance of a constraint. A score of zero marks a *hard* constraint that must always be satisfied, while higher scores mark less important *soft* constraints that should be satisfied if possible. Typically, well-formedness conditions such as agreement receive scores near zero, while mere preferences have higher scores.

For a grammar of German, the positional constraint should therefore be rewritten as follows:

```
{X:SYN} : obj_position : 0.5 :
  X.label = obj  ->  distance(X@id, X^id) < 0;
```

Here the score of 0.5 indicates that the object preferably follows the verb, but the alternate ordering is still allowed. This, however, only happens if satisfying this constraint would violate another, more important constraint. For instance, German subjects typically appear in the nominative case, while objects do not. Therefore, the more important agreement constraint could be written as follows:

```
{X:SYN} : subj_case : 0.1 :
  X.label = subj  ->  X@case = nom;
```

Together, these two constraints correctly model the case of the unmarked word order SVO (both constraints are satisfied) and of topicalized word order OVS (only the more important constraint is satisfied).

The grammar writer can choose the score of each constraint freely. For maximum flexibility, so-called dynamic constraints can carry a score that varies depending on specific characteristics of the utterance being parsed. A distance constraint, for instance, states that certain constructions prefer short attachments, as in the case of relative clauses and discontinuous verbs. Its score can be computed as follows:

$$\text{weight} = \frac{\alpha}{\alpha + \text{distance(governor, dependant)}}$$

The free parameter α determines the slope of the function. It could be set to $\alpha = 100$ for relative clauses but to $\alpha = 5$ for the distance between determiner and noun. Depending on the actual distance, this constraint may be stronger or weaker than the positional constraint mentioned above.

When a constraint applies to more than one dependency relation, it can constrain relations on different levels. The following constraint expresses the regularity that a syntactic subject usually fills the thematic role of an agent:

```
{X:SYN,Y:AGENT} : subj_is_agent : 0.5 :
  X.label = subj & X^id = Y@id  ->  Y^id = X@id;
```

This constraint simply advises that an `AGENT` relation be established between the same words as the subject relation. Since this co-ocurrence is not universal, the constraint has been made soft. (A more accurate version would also make exceptions for particular verbs where it never occurs, such as passives or stative verbs.)

Although this constraint takes the form of a logical implication (`->`), the `AGENT` relation need not be established after the subject relation. Both representations can be built independently of each other, one after another in either order, or concurrently. The solution methods detailed in section 3.3 operate concurrently on all levels of description.

A complete structural analysis on multiple levels may violate more than one soft constraint. Therefore, an aggregation function is needed which maps the set of violated constraints to an assessment of the whole structure. In WCDG, the simplest such function is employed: The scores of all violated constraints are multiplied together.[3] This aggregation function does not preserve the strict hierarchy of constraints, i. e., a violation of multiple weak constraints can very well be worse than a violation of a single more important constraint. In contrast, in a strictly hierarchical framework, such as Optimality Theory (Prince and Smolensky 1993), no stronger constraint can ever be overruled by weaker ones.

Assigning scores higher than 0 to constraints can serve two different purposes:

- By allowing linguistic rules to be broken if necessary, the grammar is made robust against input that actually breaks the rules; on ungrammatical input, processing can proceed normally except for a lower score. By looking at the constraints that are not satisfied, it can be determined how deviant the input was and what exactly was wrong about it.
- Soft constraints also allow the grammarian to postulate preferences that are not expected to be always true. Formulated as hard constraints, these rules would render many common usages extragrammatical, while as a soft constraint they do not hinder their analysis appreciably. They also serve as tie-breakers when different structures would be equally grammatical otherwise. In this way, soft constraints can help to reduce the ambiguity created by writing an overgenerating grammar in the first place.

Assigning scores to the constraints is a labor-intensive and error-prone task. Although in theory, the relative importance of constraints could be estimated by counting how often they are satisfied on an example corpus, this only works if the corpus actually contains all possible phenomena that the grammar should allow. Such a corpus of ungrammatical utterances has not been available so far.

However, it is possible to find constraint scores automatically by measuring how well a particular set of constraint scores lets the grammar perform on a given corpus. In a previous experiment, (Schröder et al. 2001), this task was successfully solved by genetic algorithms. Assuming that the grammar writer specifies only which constraints should be hard and which should be soft, random assignments and changes to the scores of soft constraints were evaluated for their performance on a given corpus, and better vectors of scores were evolved over time. Ultimately, the automatically derived vector of scores performs comparably to a vector manually selected by the grammar author when the grammar is applied to a separate test corpus.

[3] This explains the somewhat unintuitive rule that important constraints carry low scores and less important constraints carry high scores.

3.3 Parsing as constraint optimization

The robustness afforded by making constraints soft comes at a price. The structure that a WCDG postulates for an utterance is the one which has the highest possible aggregated score. The problem defined by a WCDG therefore constitutes an instance of the general constraint optimization problem, which is known to be $\mathcal{NP}$-hard. This means that in the worst case, proving a structure optimal can take time exponential in the sentence length. Depending on the size of the problem, it can be infeasible to conduct this proof. However, it is usually possible to efficiently obtain a structural analysis for any utterance that is more or less similar to the theoretical optimum. The number of search errors decreases as more effort is spent; section 5 details how processing effort can be traded against accuracy.

Various algorithms with different advantages are implemented in the WCDG parsing system to solve the optimality problem. Both complete and heuristic solution methods are available.

Systematic tree search methods successively build up larger and larger *partial* analyses until a complete solution has been found. For the grammar employed in section 4, the best-first algorithm works well on medium-sized problems (sentences up to ten words) but uses an unpredictable amount of processing time for larger sentences. Due to the systematic exploration of the search space, optimality can be guaranteed for the final result as long as sufficient computational resources are available.

(Schröder et al. 2000) reports on experiments to improve the efficiency of the systematic search for solutions. An important goal of such a search is to quickly obtain a complete solution with a score above 0. This score can then be used to disregard all partial structures with lower scores, since they cannot possibly extend to a better complete solution.

The most successful modification of a pure best-first search is to modify the ranking of intermediary solution candidates. When two partial structures with the same partial score exist, the more complete one should be investigated first, since fewer additional constraint failures are to be expected while completing it. In accordance with the multiplicative aggregation criterion, a ranking function that multiplies the square root of the length of a partial structure to its score has proven most efficient.

Since very large optimization problems are intractable for complete solution methods, heuristic search methods are usually employed. The most successful methods of heuristic search in WCDG are variants of repair-based local search. Processing starts with a complete but arbitrary analysis, which is then transformed step by step into one that has a higher score under the current grammar. Under reasonable time constraints, these methods actually achieve better results than a complete search.

In the heuristic solution method employed for the experiments in Section 4, violated constraints serve to direct the transformation. Looking at the most important violated constraint, a local search is conducted for a similar structure which repairs the error and has a higher overall score, or, failing that, as high a score as possible. When no such structure can be found without repeating a previous error, the repair fails. The method then tackles the next most important error, until it has removed or failed to remove each error. An important advantage is that a nearly correct solution is usually available long before the local search terminates. A more detailed description can be found in (Foth, Menzel, and Schröder 2000).

Another experimental variant of heuristic search uses *guided local search* to transform an analysis (Daum and Menzel 2002). To overcome the problem that local search can converge to a local rather than a global maximum, this method tries to learn features of good solution candidates and dynamically adapts the evaluation function, thus turning local optima into ascents and allowing more neighbouring problem states to be tried.

Since all heuristic solution methods use some kind of local search to find alternative structures, they cannot guarantee optimal results even if they were given unlimited processing time. In practice, however, they consistently obtain better solutions than complete search when the time *is* limited. In particular, they guarantee that a complete solution is available at any time and that the quality of their current best solution improves monotonically over time.

4 Robustness against ill-formed input

To demonstrate the extent of the architecture's robustness against ill-formed input, its performance was tested on input of varying regularity. The experiment used a medium-sized WCDG of German (around 500 constraints) developed for the Verbmobil domain (Wahlster 2000). A section with 203 utterances was selected from this corpus with an average length of 8.8 words per sentence. The transcriptions are free of phonetic ambiguity; however, the proliferation of homonymous form in German leads to an average lexical ambiguity of 3.6 forms per word. For this corpus, the heuristic solution method employed here takes an average of 96.8 seconds per sentence to terminate (with a median of 12 seconds) on a SPARC Ultra 5 processor. The corpus is analysed with an accuracy of 96.63% .

The Verbmobil corpus consists of manually transcribed spontaneous speech which contains slight grammatical errors. To test the robustness against more serious errors, the sentences were then randomly distorted further by three different methods:

1. the inflection of a word may change (50%)
2. a word may be deleted totally from the input (30%)
3. the order among two sister constituents in the input may be reversed (20%).

On the average, an error was introduced every 10 words. The annotations that represent the desired analyses of the original utterances were modified to match the resulting input. The system was then allowed a limited time to parse each of the modified utterances. To obtain different levels of ungrammaticality, this distortion was repeated 10 times and the experiment repeated at each stage.

Thus, the utterance

Dann lassen Sie uns doch noch einen Termin ausmachen
(Let us schedule another appointment then)

might be successively transformed into

- *Dann lassen Sie uns ∅ noch einen Termin ausmachen*
- *Dann* ***lasse*** *Sie uns noch einen Termin ausmachen*
- *Dann lasse Sie noch einen Termin ausmachen* ***uns***

The accuracy of the parsing system was measured by counting how many dependency relations were established correctly with respect to structure and label (i.e. the selection of an incorrect lexical variant was not considered an error). Since the solution method always returns a complete solution with a previously known number of relations, there is no difference between recall and precision.

In order to obtain a quality measure comparable to the standard PARSEVAL method, both the dependency annotations and the dependency trees produced by the parser were converted into constituency structures so that *Labelled Recall* and *Labelled Precision*, as defined in (Black et al. 1991), can be determined. Figure 2 shows that the measures do not differ significantly.

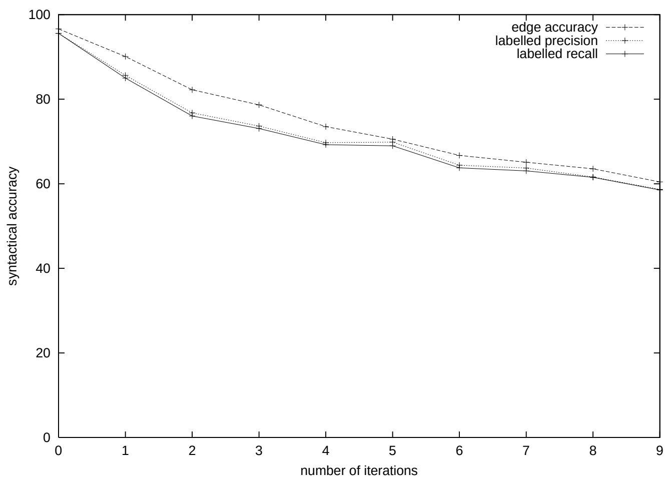

Fig. 2. Accuracy vs. error rate for randomly distorted input

It can be seen that the ability of the system to retrieve the intended syntactic structures only degrades slowly over the well-formedness of the input. The number of correctly established relations drops from 96.63% (which corresponds to the performance on the original sentences as obtained from the corpus) to 60.44% for a corpus that has been distorted ten times in succession. A totally uninformed algorithm that simply subordinates all other words under the finite verb with the most frequent label can achieve only 26% accuracy.

To determine the impact of the individual type of errors, additional experiments were carried out where only one type of error was introduced every 5 words. As expected, the reversal of constituents is relatively easy to accommodate (88.1% accuracy), since in WCDG, ordering constraints can easily be made defeasible (in contrast to phrase structure grammar). The exchange of a lexical form can (in German) introduce multiple agreement errors and is somewhat more disruptive (83.1% accuracy). The outright deletion of input is the gravest distortion (78.7% accuracy), since it can destroy the cohesiveness of an utterance.

The supporting influence of semantic information on syntactic analysis could not be conclusively demonstrated in this experiment. When the results on the distorted grammar were compared with a run where the semantic levels of analysis were not used, only an insignificant further degradation of quality was observed. This is due to the fact that the Verbmobil domain is actually not well-suited for providing semantic support. Most of the recurring verbs (have, do, make) do not establish useful selectional preferences, and most of the noun phrases are actually semantically weak pronouns. In addition, the relevant domain knowledge is usually conveyed by specific dates or times of the day, which the grammar employed does not interpret.

However, the potential for feedback from other levels back to the syntactic analysis was demonstrated earlier for another domain (Menzel and Schröder 1999). Figure 3 shows the sentence accuracy when analyzing various distortions of the sentence *Der Mann besichtigt den Marktplatz.* (The man visits the market place), which establishes a strong preference for *man* filling the agent role. Although usually the syntactic structure will determine the assignment of thematic roles of the referents even if selectional restrictions disagree, sufficient evidence from semantic constraints or domain knowledge can force the analysis to revert to the more probable reading even if the surface form is misleading. Again, the more errors are made the more likely it is that the intended proposition is not recognized.

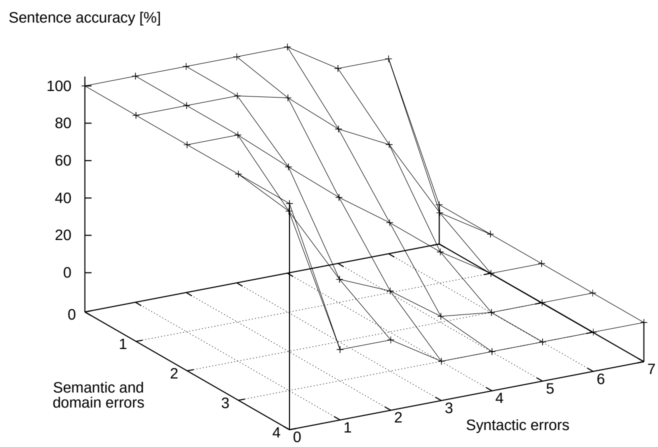

Fig. 3. Mutual compensation of deficits between syntax and semantics. Sentence accuracy is plotted against severity of syntactic distortion, and against supportive or conflicting semantic and domain knowledge.

It is one of the major advantages of WCDG that it does not merely tolerate language errors, but is also able to detect and distinguish different types of error. The violations of constraints obtained during normal processing can be directly used as diagnosis results. This ability is crucial for language teaching systems that must provide accurate error explanations. It also forms the basis for the error-driven transformational solution methods (cf. Section 3.3).

Another important indication for the inherent robustness of a WCDG parser is its ability to deal with fragmentary input. Since a WCDG need not impose the condition that analyses must form a cohesive and consistent tree, missing input forms may lead to constraint violations or may even cause the structure to break into parts. In particular, we have been able to show (Foth et al. 2000) that an arbitrary prefix of a sentence can usually be analysed into a meaningful partial analysis, which can be modified as more word forms need to be considered.

Since constraints are defined on edges rather than complete constituents, such a kind of prefix parsing provides for an extremely eager form of incremental processing where every incoming word form is immediately integrated into the already existing structure without the necessity to make any assumptions about the not yet observed input. Instead, particular constraint violations signal missing or incoherent syntactic material and can easily be interpreted as expectations about upcoming parts of the utterance. For instance, in the incomplete utterance *Dann treffen wir* (Then let's meet), German grammar requires a direct object, and this requirement leads to the only serious constraint violation.

5 Robustness against temporal pressure

Intuitively, a time-adaptive algorithm needs some leeway in its processing strategies in order to be able to speed up the computation when this seems desirable. The first experiment is supposed to verify whether our formalism possesses this flexibility. Therefore, a pre-defined and static temporal pressure has been applied to the system.

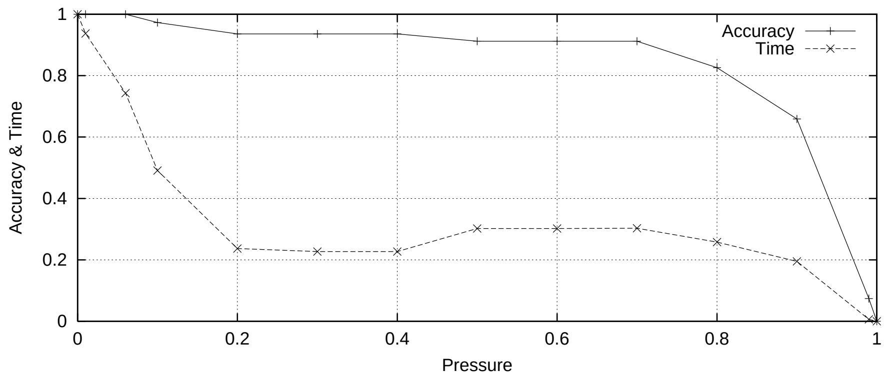

Fig. 4. Behavior of best-first search under static temporal pressure

Figure 4 (taken from (Schröder 2002)) shows the outcome of parsing the reference corpus using tree search where more and more soft constraints have been made hard (and thus more effective) to simulate temporal pressure. Both the parsing accuracy and the time requirement are compared to the baseline case (no temporal pressure).

The rationale behind this experiment is that the stronger a constraint is, the more exceptional the cases are where an optimal solution violates the constraint under consideration. For a given time pressure p, all constraint weights smaller than (or equal to) p have been set to zero. This has two consequences:

- In those exceptional cases where the optimal solution indeed violates such a constraint, at best a different structure is selected or the sentence becomes completely unparsable. As a result, the accuracy of the system drops.
- Since the now hard constraints eliminate some solution candidates once and for all early during the analysis, a speedup in processing is to be expected.

Beyond this expected behavior another observation can be made from Figure 4: The degradation of the quality, i. e., accuracy in this case, is much weaker and delayed compared to the speedup. This graceful response is a prerequisite for time-adaptive behavior, since it allows adjusting the pressure according to the available time.

Figure 5 presents the sentence accuracy for a similar experiment where the transformation-based local search method has been used. The time pressure p translates into an internal threshold which determines how far a local search path is explored. The experiments confirm that both solution methods are robust and flexible enough to be used under temporal pressure.

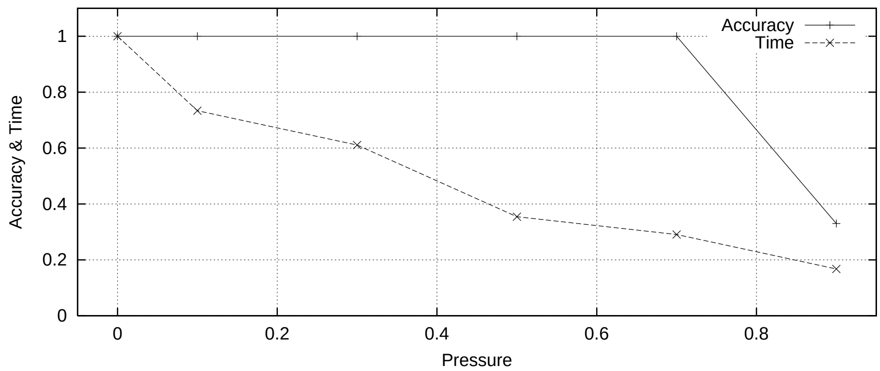

Fig. 5. Behavior of transformational search under static temporal pressure

However, transformation-based methods are even more attractive from this perspective. Since a solution candidate is available from the very beginning, they can be interrupted at an arbitray point in time during the analysis. Thus, even under strong temporal pressure a solution is obtained which at least has the potential to be improved if more processing time is spent.

Figure 6 shows how the accuracy of an intermediate solution evolves over processing time in an example run. Although not every repair step actually increases the accuracy of analysis, an approximation of the optimal solution is quickly found. Building the exact desired structure takes a longer time.

Both kinds of search methods could be tuned to a more sophisticated behavior during the parsing process itself, by adjusting the pressure imposed according to the

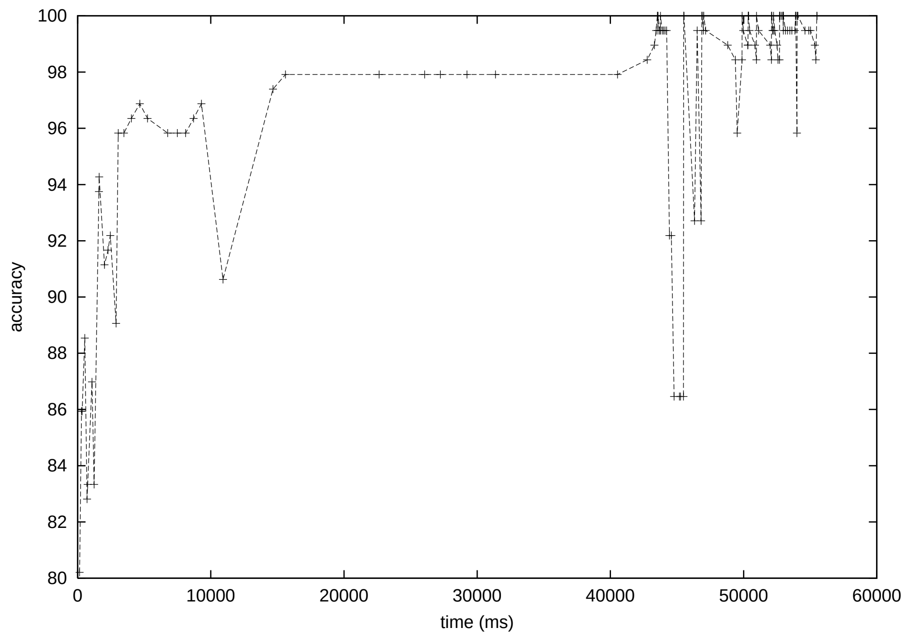

Fig. 6. Performance profile for a problem instance of transformation-based search

time ressources left and the result achieved so far. Thus, also a completely dynamic time-adaptive behavior seems feasible.

6 Related work

There has been an abundance of attempts to increase the robustness of parsers against ill-formed and otherwise unexpected input, which go back more than twenty years. For a comprehensive overview of these efforts see (Stede 1992). Most of them are based on the idea of post-hoc fitting procedures, which, after a parse failure occured, modify the general parsing mechanism in a way that establishes the conformity with the grammar. Usually the parser continues by ignoring, adding, substituting or rearranging parts of the input sentence until a minimal set of modifications has been determined which are necessary to successfully parse the sentence.

In contrast to these approaches, the system described in this paper does not require a special treatment of error cases as opposed to normal ones. Since constraint violations are an important means of guiding the solution procedure anyhow, they are observed in arbitrary utterances including the well-formed ones. Therefore, ill-formed or extragrammatical, as well as well-formed input are all handled by the same computational mechanism and the two important cases differ only in the weights of violated constraints.

More recently, stochastic approaches to parsing, e. g. (Collins 1999), have been highly successful from the perspective of processing unrestricted text. Their ability for analysing ill-formed input, however, has only been investigated for the case of speech recognition results, e. g. (Roark 2001). These results show that due to its inherent robustness a parser trained on non-distorted data can also be used to process real speech recognition results without exhibiting a particular bias towards non-distorted utterances as found in the corpus. However, being trained directly on transcribed spontaneous speech the parser can certainly not be expected to derive a normative model of flawless utterances from that kind of data, nor can it provide the ability to explain observed deviations from the norm as it is required in other applications like foreign language tutoring systems, c. f. (Rosé and Lavie 2001).

The problem of temporal adaptivity has attracted fairly little attention in the parsing literature so far. This may well be explained by the fact that the real benefit of graceful degrading under temporal pressure only becomes obvious in strictly incremental speech processing, where keeping pace with the incoming data is crucial for a successful understanding. For the time being, however, spoken language understanding poses more serious problems to be solved.

While there is a number of results concerning the general relationship between output quality and processing time available, no alternative approaches to devise an actually interruptable algorithm for natural language parsing have been found in the literature. This particular focus to the quality/efficiency issue is caused by the widespread use of parsing architectures based on beam search, which can be easily tuned by manipulating their beam width.

Experiments with different beam sizes have been reported for stochastic approaches trained on treebank data, as well as for systems with traditional handcrafted grammars. Narrowing down the beam size of a stochastic parser based on a bi-lexical Markovian grammar model Collins (1999) achieved a speedup by a factor of 2.45 while reducing labelled recall and precision (relative) by only 1.65% and 1.41% respectively on the standard Penn Treebank setting. For his lexicalized top-down parser Roark (2001) even reports a speedup of one order of magnitude if a relative loss in parsing accuracy of 2.55% for labelled recall and 2.65% for labelled precision can be tolerated.

Rosé and Lavie (2001) extended a generalized LR Tomita parser, as well as a left-corner chart parser for an LFG-like unification formalism to enhance their robustness. The additional mechanisms include word skipping, non-terminal insertion, and a flexible unification. A beam search is carried out to determine the parse which can be found with a minimum of modifications applied to the input string. Again, the parser shows the typical antagonistic relationship between speedup and quality reduction. Tested with different beam widths on a translation task for spontaneous speech the proportion of acceptable translations went down e. g. from 69.67% to 68.50% for the GLR parser as the speed increased by a factor of two approximately. Similar results have also been obtained by limiting the maximum number of skipped words per sentence.

Several parsers for the German part of the Verbmobil corpus have been developed elsewhere among them a HPSG-based (Kiefer, Krieger, and Nederhof 2000) and a probabilistic one (Ruland 2000). Unfortunately, evaluation results are available only for the latter one. Being trained on 19,750 sentences and tested on 1000 sentences from a German treebank, the system achieved a labelled recall of 83.5% and a labelled precision of 90.2%. Note, however, that the experimental conditions under which these results have been obtained are not the same as those used for the experiments reported above.

The idea of determining dependency structures for natural language utterances by means of constraint satisfaction techniques goes back to (Maruyama 1990) who proposed to dynamically reinforce the disambiguating power of a grammar according to the actual processing needs by successively adding constraints until full disambiguation can been achieved. (Menzel 1995) extended the approach by introducing weighted constraints to improve robustness and achieve full disambiguation in (almost) any case.

A closely related framework is Constraint Grammar (Karlsson et al. 1995). Here, syntactic parsing ist understood as a kind of tagging procedure which assigns functional categories to word forms but leaves the exact attachment point partly underspecified. Robustness is increased by means of heuristic rules which are only invoked if full disambiguation cannot be achieved by means of the normal ones. On a randomly sampled test set of 4180 word forms from the Brown corpus a recall of 96.7% has been reported and a precision of 77.5% can be calculated from the available numbers, indicating a relatively high level of unresolved ambiguity.

7 Conclusions

An architecture for robust parsing has been proposed which is based on a clear separation between possible structural descriptions and correctness conditions. Correctness conditions are expressed as weighted constraints over acceptable structural relationships and parsing amounts to determining an optimal interpretation in a procedure of structural disambiguation. Due to the separation between potential structures and defeasible well-formedness constraints, a system could be designed which maintains the notion of a normative standard but allows the treatment of deviations from the norm.

Disambiguation is carried out on several description levels in parallel, where structures on different levels are mapped onto each other, again by means of weighted constraints. Such a multi-level architecture provides for

- a simultaneous consideration of possibly contradicting evidence from a broad range of information sources,
- the computation of rich structural representations integrating aspects of syntax, semantics, reference and information structure, and
- a mutual compensation of partial information deficits on the different representational levels.

Because evidence arbitration through global optimization is the general computational mechanism of the system, inconsistencies are no longer treated as an exception but as the typical case in language processing. This design is a significant extension over pure constraint satisfaction approaches as described in (Maruyama 1990), (Harper et al. 1995), (Duchier 1999) or (Blache 2000). It not only allows full disambiguation of an utterance in most cases, but also facilitates an easy integration of preferential and uncertain information. Moreover, inconsistencies in a solution can be evaluated and diagnosed in a way which allows the diagnostic findings to be used as guiding hints for subsequent analysis steps, as expectations of the not yet observed right context or as error explanations for students of a foreign language.

The experiments conducted so far indicate that the system is able to achieve a comparatively high degree of robustness against ill-formed and incomplete utterances. Although in some cases strange interpretations can be obtained, usually there is no complete failure. Most importantly, robustness has not been established as an add-on feature but is directly derived from the properties of the fundamental processing scheme. As a consequence, the approach treats robustness as a multi-dimensional phenomenon, integrating robustness against ungrammaticality, incomplete utterances, insufficient competence and temporal pressure.

Further investigations will need to show that the general robust behaviour can also be achieved on larger corpora. This will require to include (perhaps uncertain) evidence from external components of shallow processing, like taggers or chunkers. Moreover, to cope with the complexity of long sentences, more advanced techniques for incremental left-to-right processing and partial parsing will need to be developed.

Another promising direction might be the investigation of the potential for a relevance-driven parsing. By biasing a weighted constraint dependency grammar with knowledge about a particular information-seeking task, the parser might eventually be put in a position to distinguish between relevant and irrelevant information contributions within a text. Eventually this might lead to a language processing architecture which is able to adapt the structural depth of its analyses according to the given data. Such a result would not only be a major step towards bridging the current gap between shallow and deep approaches to parsing, but could also open up new application opportunities in the area of information extraction.

Acknowledgements

The authors are particularly grateful to one of the anonymous reviewers who carefully pinpointed many shortcomings of an earlier version of this paper.

The research reported in this paper was partially supported by a grant from the German Research Council (DFG) under contract Me 1472/1-2.

References

Abney, Steven. (1991) Parsing by chunks. In R. Berwick, S. Abney and C. Tenny (eds.), *Principle-based Parsing*. Dordrecht: Kluwer Academic Publishers.

Black, Ezra, Steven Abney, Dan Flickenger, Claudia Gdaniec, Ralph Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith Klavans, Mark Liberman, Mitchell P. Marcus, Salim Roukos, Beatrice Santorini and Tomek Strzalkowski. (1991) A Procedure for Quantitatively Comparing the Syntactic Coverage of English. In *Proceedings DARPA Speech and Natural Language Workshop*. pp. 306–311. Pacific Grove, CA.

Blache, Philippe. (2000) Constraints, linguistic theories and natural language processing.

In D. Christodoulakis (ed.) *Natural Language Processing.* Lecture Notes in Artificial Intelligence. Berlin: Springer-Verlag.

Boddy, Mark and Thomas L. Dean. (1994) Deliberation scheduling for problem solving in time-constrained environments. *Artificial Intelligence*, **2**: 245–285.

Charniak, Eugene. (2000) A maximum-entropy-inspired parser. In *Proc. Language Technology Joint Conference ANLP-NAACL*, pp. 132–139. Seattle, WA.

Charniak, Eugene, Glenn Carroll, John Adcock, Anthony Cassandra, Yoshihiko Gotoh, Jeremy Katz, Michael Littman and John McCann. (1996) Taggers for parsers. *Artificial Intelligence*, **1-2**: 45–57.

Church, Kenneth W. (1988) A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the 2nd Conference on Applied Natural Language Processing.* pp. 136–143. Austin, TX.

Collins, Michael J. (1999) *Head-Driven Statistical Models for Natural Language Parsing.* Ph.D. thesis, University of Pennsylvania, Philadelphia.

Cover, Thomas M. and Joy A. Thomas. (1991) *Elements of Information Theory.* New York: Wiley.

Crain, S. and M. Steedman. (1985) On not being led up the garden path. In D. R. Dowty, L. Kartunen, and A. M. Zwicky (eds.) *Natural Language Parsing: Psychological, computational, and theoretical perspectives.* pp. 320–358. Cambridge: Cambridge University Press.

Daum, Michael and Wolfgang Menzel. (2002) Parsing Natural Language using Guided Local Search. In *Proc. 15th European Conference on Artificial Intelligence, ECAI-2002.* pp. 435–439. Lyon, France.

Duchier, Denys. (1999) Axiomatizing dependency parsing using set constraints. In *Proc. 6th Meeting on the Mathematics of Language.* Orlando, FL.

Foth, Kilian, Wolfgang Menzel, Horia F. Pop and Ingo Schröder. (2000) An experiment in incremental parsing using weighted constraints. In *Proceedings 18th International Conference on Computational Linguistics, Coling 2000.* pp. 1026-1030, Saarbrücken, Germany.

Foth, Kilian, Wolfgang Menzel and Ingo Schröder. (2000) A transformation-based parsing technique with anytime property. In *Proceedings of the International Workshop on Parsing Technologies (IWPT-2000).* pp. 89–100. Trento, Italy.

Goodman, Joshua. (1996) Parsing algorithms and metrics. In *Proceedings 34th Annual Meeting of the ACL.* pp. 177-183. Santa Cruz, USA.

Harper, Mary P., Randall A. Helzermann, C. B. Zoltowski, B. L. Yeo, Y. Chan, T. Steward and B. L. Pellom. (1995) Implementation issues in the development of the PARSEC parser. *Software - Practice and Experience*, **8**: 831–862.

Harper, Mary P., Leah H. Jamieson, Carla B. Zoltowski and Randall A. Helzerman. (1993) Semantics and constraint parsing of word graphs. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.* pp. 63–66. Minneapolis, MN.

Heinecke, Johannes, Jürgen Kunze, Wolfgang Menzel and Ingo Schröder. (1998) Eliminative parsing with graded constraints. In *Proceedings 17th International Conference on Computational Linguistics, 36th Annual Meeting of the ACL, Coling-ACL '98.* pp. 526–530. Montreal, Canada.

Karlsson, Fred, Atro Voutilainen, Juha Heikkilä and Arto Anttila. (1995) Constraint Grammar – A Language-Independent System for Parsing Unrestricted Text. Berlin, New York: Mouton de Gruyter.

Kiefer, Bernd, Ulrich Krieger and Mark-Jan Nederhof. (2000) Efficient and robust parsing of word hypotheses graphs. In Wolfgang Wahlster (ed.) *Verbmobil: Foundations of Speech-to-Speech Translation.* pp. 280–295. Berlin: Springer-Verlag.

Maruyama, Hiroshi. (1990) Structural disambiguation with constraint propagation. In *Proceedings 28th Annual Meeting of the ACL*, pp. 31–38. Pittburgh, PA.

Massaro, Dominic W. (1998) *Perceiving Talking Faces: From Speech Perception to a Behavioral Principle.* Cambridge, MA: MIT Press.

Menzel, Wolfgang. (1995) Robust Parsing of Natural Language. in *KI-95: Advances in Artificial Intelligence*, pp. 19–34. Berlin: Springer-Verlag.

Menzel, Wolfgang and Ingo Schröder. (1999) Error diagnosis for language learning systems. *ReCALL*, (special edition): 20–30.

Pollard, Carl and Ivan A. Sag. (1994) *Head-Driven Phrase Structure Grammar.* Chicago: The University of Chicago Press.

Prince, Alan and Paul Smolensky. (1993) Optimality theory: Constraint interaction in generative grammar. Technical Report 2, Center for Cognitive Science, Rutgers University.

Roark, Brian Edward. (2001) *Robust Probabilistic Predictive Syntactic Processing: Motivation, Models, and Applications.* Ph.D. thesis, Department of Cognitive and Linguistic Sciences, Brown University.

Rosé, Carolyn Penstein and Alon Lavie. (2001) Balancing robustness and efficiency in unification-augmented context-free parsers for large practical applications. In G. van Noord and P. Junqua (eds.) *Robustness in Language and Speech.* Dordrecht: Kluwer.

Ruland, Tobias. (2000) Probabilistic LR-parsing with symbolic processing. In Wolfgang Wahlster (ed.) *Verbmobil: Foundations of Speech-to-Speech Translation.* pp. 147–162. Berlin: Springer-Verlag.

Schröder, Ingo. (2002) *An Anytime Approach to Natural Language Parsing Using Constraint Satisfaction.* Ph.D. thesis, Universität Hamburg.

Schröder, Ingo, Wolfgang Menzel, Kilian Foth and Michael Schulz. (2000) Modeling dependency grammar with restricted constraints. *Traitement automatique des langues (T.A.L.)*, **1**: 113–144.

Schröder, Ingo, Horia F. Pop, Wolfgang Menzel and Kilian A. Foth. (2001) Learning Grammar Weights Using Genetic Algorithms. In *Proceedings Recent Advances in Natural Language Processing, RANLP-2001.* pp. 235–239. Tsigov Chark, Bulgaria.

Stede, Manfred. (1992) The search for robustness in natural language understanding. *Artificial Intelligence Review* **4**: 383–414.

Tesnière, Lucien. (1959) Eléments de syntaxe structurale. Paris: Klincksieck.

Trueswell, J. C., M. K. Tanenhaus, and S. M. Garnsey. (1994) Semantic influences on parsing: Use of thematic roles information in syntactic ambiguity resolution. *Journal of Memory and Language* **33**: 285–318.

Wahlster, Wolfgang (ed). (2000) *Verbmobil: Foundations of Speech-to-Speech Translation.* Berlin: Springer-Verlag.