Hybrid Parsing:
Using Probabilistic Models as Predictors for a Symbolic Parser

Kilian A. Foth, Wolfgang Menzel
Department of Informatics
Universität Hamburg, Germany
{foth|menzel}@informatik.uni-hamburg.de

Abstract

In this paper we investigate the benefit of stochastic predictor components for the parsing quality which can be obtained with a rule-based dependency grammar. By including a chunker, a supertagger, a PP attacher, and a fast probabilistic parser we were able to improve upon the baseline by 3.2%, bringing the overall labelled accuracy to 91.1% on the German NEGRA corpus. We attribute the successful integration to the ability of the underlying grammar model to combine uncertain evidence in a soft manner, thus avoiding the problem of error propagation.

1 Introduction

There seems to be an upper limit for the level of quality that can be achieved by a parser if it is confined to information drawn from a single source. Stochastic parsers for English trained on the Penn Treebank have peaked their performance around 90% (Charniak, 2000). Parsing of German seems to be even harder and parsers trained on the NEGRA corpus or an enriched version of it still perform considerably worse. On the other hand, a great number of shallow components like taggers, chunkers, supertaggers, as well as general or specialized attachment predictors have been developed that might provide additional information to further improve the quality of a parser's output, as long as their contributions are in some sense complementory. Despite these prospects, such possibilities have rarely been investigated so far.

To estimate the degree to which the desired synergy between heterogeneous knowledge sources can be achieved, we have established an experimental framework for syntactic analysis which allows us to plug in a wide variety of external predictor components, and to integrate their contributions as additional evidence in the general decision-making on the optimal structural interpretation. We refer to this approach as hybrid parsing because it combines different kinds of linguistic models, which have been acquired in totally different ways, ranging from manually compiled rule sets to statistically trained components.

In this paper we investigate the benefit of external predictor components for the parsing quality which can be obtained with a rule-based grammar. For that purpose we trained a range of predictor components and integrated their output into the parser by means of soft constraints. Accordingly, the goal of our research was not to extensively optimize the predictor components themselves, but to quantify their contribution to the overall parsing quality. The results of these experiments not only lead to a better understanding of the utility of the different knowledge sources, but also allow us to derive empirically based priorities for further improving them. We are able to show that the potential of WCDG for information fusion is strong enough to accomodate even rather unreliable information from a wide range of predictor components. Using this potential we were able to reach a quality level for dependency parsing German which is unprecendented so far.

2 Hybrid Parsing

A hybridization seems advantageous even among purely stochastic models. Depending on their degree of sophistication, they can and must be trained on quite different kinds of data collections, which due to the necessary annotation effort are available in vastly different amounts: While training a probabilistic parser or a supertagger usually

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 321–328,
Sydney, July 2006. ©2006 Association for Computational Linguistics

requires a fully developed tree bank, in the case of taggers or chunkers a much more shallow and less expensive annotation suffices. Using a set of rather simple heuristics, a PP-attacher can even be trained on huge amounts of plain text.

Another reason for considering hybrid approaches is the influence that contextual factors might exert on the process of determining the most plausible sentence interpretation. Since this influence is dynamically changing with the environment, it can hardly be captured from available corpus data at all. To gain a benefit from such contextual cues, e.g. in a dialogue system, requires to integrate yet another kind of external information.

Unfortunately, stochastic predictor components are usually not perfect, at best producing preferences and guiding hints instead of reliable certainties. Integrating a number of them into a single systems poses the problem of error propagation. Whenever one component decides on the input of another, the subsequent one will most probably fail whenever the decision was wrong; if not, the erroneous information was not crucial anyhow. Dubey (2005) reported how serious this problem can be when he coupled a tagger with a subsequent parser, and noted that tagging errors are by far the most important source of parsing errors.

As soon as more than two components are involved, the combination of different error sources migth easily lead to a substantial decrease of the overall quality instead of achieving the desired synergy. Moreover, the likelihood of conflicting contributions will rise tremendously the more predictor components are involved. Therefore, it is far from obvious that additional information always helps. Certainly, a processing regime is needed which can deal with conflicting information by taking its reliability (or relative strength) into account. Such a preference-based decision procedure would then allow stronger valued evidence to override weaker one.

3 WCDG

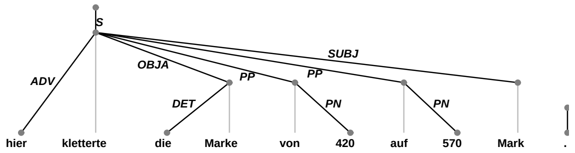
An architecture which fulfills this requirement is *Weighted Constraint Dependency Grammar*, which was based on a model originally proposed by Maruyama (1990) and later extended with weights (Schröder, 2002). A WCDG models natural language as *labelled dependency trees* on words, with no intermediate constituents assumed. It is entirely *declarative*: it only contains rules (called *constraints*) that explicitly describe the properties of well-formed trees, but no derivation rules. For instance, a constraint can state that determiners must precede their regents, or that there cannot be two determiners for the same regent, or that a determiner and its regent must agree in number, or that a countable noun must have a determiner. Further details can be found in (Foth, 2004). There is only a trivial generator component which enumerates all possible combinations of labelled word-to-word subordinations; among these any combination that satisfies the constraints is considered a correct analysis.

Constraints on trees can be *hard* or *soft*. Of the examples above, the first two should probably be considered hard, but the last two could be made defeasible, particularly if a robust coverage of potentially faulty input is desired. When two alternative analyses of the same input violate different constraints, the one that satisfies the more important constraint should be preferred. WCDG ensures this by assigning every analysis a score that is the product of the weights of all instances of constraint failures. Parsing tries to retrieve the analysis with the highest score.

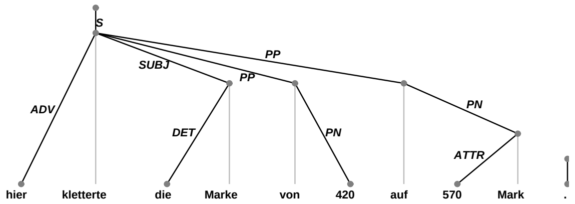
The weight of a constraint is usually determined by the grammar writer as it is formulated. Rules whose violation would produce nonsensical structures are usually made hard, while rules that enforce preferred but not required properties receive less weight. Obviously this classification depends on the purpose of a parsing system; a prescriptive language definition would enforce grammatical principles such as agreement with hard constraints, while a robust grammar must allow violations but disprefer them via soft constraints. In practice, the precise weight of a constraint is not particularly important as long as the relative importance of two rules is clearly reflected in their weights (for instance, a misinflected determiner is a language error, but probably a less severe one than duplicate determiners). There have been attempts to compute the weights of a WCDG automatically by observing which weight vectors perform best on a given corpus (Schröder et al., 2001), but weights computed completely automatically failed to improve on the original, hand-scored grammar.

Weighted constraints provide an ideal interface to integrate arbitrary predictor components in a soft manner. Thus, external predictions are treated the same way as grammar-internal preferences, e.g. on word order or distance. In contrast to a filtering approach such a strong integration does not blindly rely on the available predictions but is able to question them as long as there is strong enough combined evidence from the grammar and the other predictor components.

For our investigations, we used the reference implementation of WCDG available from `http://nats-www.informatik.uni-hamburg.de/download`, which allows constraints to express any formalizable property of a dependency tree. This great expressiveness has the disadvantage that the parsing problem becomes $\mathcal{NP}$-complete and cannot be solved efficiently. However, good success has been achieved with transformation-based solution methods that start out with an educated guess about the optimal tree and use constraint failures as cues where to change labels, subordinations, or lexical readings. As an example we show intermediate and final analyses of a sentence from our test set (negra-s18959): 'Hier kletterte die Marke von 420 auf 570 Mark.' (*Here the figure rose from 420 to 570 DM*).



In the first analysis, subject and object relations are analysed wrongly, and the noun phrase '570 Mark' has not been recognized. The analysis is imperfect because the common noun 'Mark' lacks a Determiner.



The final analysis correctly takes '570 Mark' as the kernel of the last preposition, and 'Marke' as the subject. Altogether, three dependency edges had to be changed to arrive at this solution.

Figure 1 shows the pseudocode of the best solution algorithm for WCDG described so far (Foth et al., 2000). Although it cannot guarantee to find the best solution to the constraint satisfaction problem, it requires only limited space and can be interrupted at any time and still returns a solution. If not interrupted, the algorithm terminates when

```
$A$ := the set of levels of analysis
$W$:= the set of all lexical readings of words in the sentence
$L$ := the set of defined dependency labels
$E := A \times W \times W \times L$ = the base set of dependency edges
$D := A \times W$ = the set of domains $d_{a,w}$ of all constraint variables
$B := \emptyset$ = the best analysis found
$C := \emptyset$ = the current analysis

{ Create the search space. }
for $e \in E$
    if $\text{eval}(e) > 0$
    then $d_{a,w} := d_{a,w} \cup \{e\}$

{ Build initial analysis. }
for $d_{a,w} \in D$
    $e_0 = \arg\max_{e \in d_{a,w}} \text{score}(C \cup \{e\})$
    $C := C \cup \{e_0\}$
$B := C$
$T := \emptyset$ = tabu set of conflicts removed so far.
$U := \emptyset$ = set of unremovable conflicts.
$i :=$ the penalty threshold above which conflicts are ignored.
$n := 0$

{ Remove conflicts. }
while $\exists\, c \in \text{eval}(C) \setminus U : \text{penalty}(c) > i$
    and no interruption occurred

    { Determine which conflict to resolve. }
    $c_n := \arg\max_{c \in \text{eval}(C) \setminus U} \text{penalty}(c)$
    $T := T \cup \{c\}$

    { Find the best resolution set. }
    $R_n := \arg\max_{R \in \bigtimes \text{domains}(c_n)} \text{score}(\text{replace}(C, R))$
          where replace$(C, R)$ does not cause any $c \in T$
          and $|R \setminus C| <= 2$

    if no $R_n$ can be found

        { Consider $c_0$ unremovable. }
        $n := 0, C := B, T := \emptyset, U := U \cup \{c_0\}$
    else

        { Take a step. }
        $n := n + 1, C := \text{replace}(C, R_n)$
        if $\text{score}(C) > \text{score}(B)$
            $n := 0, B := C, T := \emptyset, U := U \cap \text{eval}(C)$

return $B$
```

Figure 1: Basic algorithm for heuristic transformational search.

no constraints with a weight less than a predefined threshold are violated. In contrast, a complete search usually requires more time and space than available, and often fails to return a usable result at all. All experiments described in this paper were conducted with the transformational search.

For our investigation we use a comprehensive grammar of German expressed in about 1,000 constraints (Foth et al., 2005). It is intended to cover modern German completely and to be robust against many kinds of language error. A large WCDG such as this that is written entirely by hand can describe natural language with great precision, but at the price of very great effort for the grammar writer. Also, because many incorrect analyses are allowed, the space of possible trees becomes even larger than it would be for a prescriptive grammar.

4 Predictor components

Many rules of a language have the character of general preferences so weak that they are easily overlooked even by a language expert; for instance, the ordering of elements in the German *mittelfeld* is subject to several types of preference rules. Other regularities depend crucially on the lexical identity of the words concerned; modelling these fully would require the writing of a specific constraint for each word, which is all but infeasible. Empirically obtained information about the behaviour of a language would be welcome in such cases where manual constraints are not obvious or would require too much effort. This has already been demonstrated for the case of part-of-speech tagging: because contextual cues are very effective in determining the categories of ambiguous words, purely stochastical models can achieve a high accuracy. (Hagenström and Foth, 2002) show that the TnT tagger (Brants, 2000) can be profitably integrated into WCDG parsing: A constraint that prefers analyses which conform to TnT's category predictions can greatly reduce the number of spurious readings of lexically ambiguous words. Due to the soft integration of the tagger, though, the parser is not forced to accept its predictions unchallenged, but can override them if the wider syntactic context suggests this. In our experiments (line 1 in Table 1) this happens 75 times; 52 of these cases were actual errors committed by the tagger. These advantages taken together made the tagger the by far most valuable information source, whithout which the analysis of arbitrary input would not be feasible at all. Therefore, we use this component (POS) in all subsequent experiments.

Starting from this observation, we extended the idea to integrate several other external components that predict particular aspects of syntax analyses. Where possible, we re-used publicly available components to make the predictions rather than construct the best predictors possible; it is likely that better predictors could be found, but components 'off the shelf' or written in the simplest workable way proved enough to demonstrate a positive benefit of the technique in each case.

For the task of predicting the boundaries of major constituents in a sentence (chunk parsing, CP), we used the decision tree model TreeTagger (Schmid, 1994), which was trained on articles from *Stuttgarter Zeitung*. The noun, verb and prepositional chunk boundaries that it predicts are fed into a constraint which requires all chunk heads to be attached outside the current chunk, and all other words within it. Obviously such information can greatly reduce the number of structural alternatives that have to be considered during parsing. On our test set, the TreeTagger achieves a precision of 88.0% and a recall of 89.5%.

Models for category disambiguation can easily be extended to predict not only the syntactic category, but also the local syntactic environment of each word (supertagging). Supertags have been successfully applied to guide parsing in symbolic frameworks such as Lexicalised Tree-Adjoning grammar (Bangalore and Joshi, 1999). To obtain and evaluate supertag predictions, we re-trained the TnT Tagger on the combined NEGRA and TIGER treebanks (1997; 2002). Putting aside the standard NEGRA test set, this amounts to 59,622 sentences with 1,032,091 words as training data. For each word in the training set, the local context was extracted and encoded into a linear representation. The output of the retrained TnT then predicts the label of each word, whether it follows or precedes its regent, and what other types of relations are found below it. Each of these predictions is fed into a constraint which weakly prefers dependencies that do not violate the respective prediction (ST). Due to the high number of 12947 supertags in the maximally detailed model, the accuracy of the supertagger for complete supertags is as low as 67.6%. Considering that a detailed supertag corresponds to several distinct predictions (about label, direction etc.), it might be more appropriate to measure the average accuracy of these distinct predictions; by this measure, the individual predictions of the supertagger are 84.5% accurate; see (Foth et al., 2006) for details.

As with many parsers, the attachment of prepositions poses a particular problem for the base WCDG of German, because it is depends largely upon lexicalized information that is not widely used in its constraints. However, such information

Predictors	Reannotated Dependencies	Transformed Dependencies
1: POS only	89.7%/87.9%	88.3%/85.6%
2: POS+CP	90.2%/88.4%	88.7%/86.0%
3: POS+PP	90.9%/89.1%	89.6%/86.8%
4: POS+ST	92.1%/90.7%	90.7%/88.5%
5: POS+SR	91.4%/90.0%	90.0%/87.7%
6: POS+PP+SR	91.6%/90.2%	90.1%/87.8%
7: POS+ST+SR	92.3%/90.9%	90.8%/88.8%
8: POS+ST+PP	92.1%/90.7%	90.7%/88.5%
9: all five	92.5%/91.1%	91.0%/89.0%

Table 1: Structural/labelled parsing accuracy with various predictor components.

can be automatically extracted from large corpora of trees or even raw text: prepositions that tend to occur in the vicinity of specific nouns or verbs more often than chance would suggest can be assumed to modify those words preferentially (Volk, 2002).

A simple probabilistic model of PP attachment (PP) was used that counts only the occurrences of prepositions and potential attachment words (ignoring the information in the kernel noun of the PP). It was trained on both the available tree banks and on 295,000,000 words of raw text drawn from the taz corpus of German newspaper text. When used to predict the probability of the possible regents of each preposition in each sentence, it achieved an accuracy of 79.4% and 78.3%, respectively (see (Foth and Menzel, 2006) for details). The predictions were integrated into the grammar by another constraint which disprefers all possible regents to the corresponding degree (except for the predicted regent, which is not penalized at all).

Finally, we used a full dependency parser in order to obtain structural predictions for *all* words, and not merely for chunk heads or prepositions. We constructed a probabilistic shift-reduce parser (SR) for labelled dependency trees using the model described by (Nivre, 2003): from all available dependency trees, we reconstructed the series of parse actions (shift, reduce and attach) that would have constructed the tree, and then trained a simple maximum-likelihood model that predicts parse actions based on features of the current state such as the categories of the current and following words, the environment of the top stack word constructed so far, and the distance between the top word and the next word. This oracle parser achieves a structural and labelled accuracy of 84.8%/80.5% on the test set but can only predict projective dependency trees, which causes problems with about 1% of the edges in the 125,000 dependency trees used for training; in the interest of simplicity we did not address this issue specially, instead relying on the ability of the WCDG parser to robustly integrate even predictions which are wrong by definition.

5 Evaluation

Since the WCDG parser never fails on typical treebank sentences, and always delivers an analysis that contains exactly one subordination for each word, the common measures of precision, recall and f-score all coincide; all three are summarized as *accuracy* here. We measure the *structural* (i.e. unlabelled) accuracy as the ratio of correctly attached words to all words; the *labelled* accuracy counts only those words that have the correct regent and also bear the correct label. For comparison with previous work, we used the next-to-last 1,000 sentences of the NEGRA corpus as our test set. Table 1 shows the accuracy obtained.[1]

The gold standard used for evaluation was derived from the annotations of the NEGRA treebank (version 2.0) in a semi-automatic procedure. First, the NEGRA phrase structures were automatically transformed to dependency trees with the DEPSY tool (Daum et al., 2004). However, before the parsing experiments, the results were manually corrected to (1) take care of systematic inconsistencies between the NEGRA annotations and the WCDG annotations (e.g. for non-projectivities, which in our case are used only if necessary for an ambiguity free attachment of verbal arguments, relative clauses and coordinations, but not for other types of adjuncts) and (2) to remove inconsistencies with NEGRAs own annotation guidelines (e.g. with regard to elliptical and co-ordinated structures, adverbs and subordinated main clauses.) To illustrate the consequences of these corrections we report in Table 1 both kinds of results: those obtained on our WCDG-conform annotations (reannotated) and the others on the raw output of the automatic conversion (trans-

[1] Note that the POS model employed by TnT was trained on the entire NEGRA corpus, so that there is an overlap between the training set of TnT and the test set of the parser. However, control experiments showed that a POS model trained on the NEGRA and TIGER treebanks minus the test set results in the same parsing accuracy, and in fact slightly better POS accuracy. All other statistical predictors were trained on data disjunct from the test set.

formed), although the latter ones introduce a systematic mismatch between the gold standard and the design principles of the grammar.

The experiments 2–5 show the effect of adding the POS tagger and one of the other predictor components to the parser. The chunk parser yields only a slight improvement of about 0.5% accuracy; this is most probably because the baseline parser (line 1) does not make very many mistakes at this level anyway. For instance, the relation type with the highest error rate is prepositional attachment, about which the chunk parser makes no predictions at all. In fact, the benefit of the PP component alone (line 3) is much larger even though it predicts *only* the regents of prepositions. The two other components make predictions about all types of relations, and yield even bigger benefits.

When more than one other predictor is added to the grammar, the beneft is generally higher than that of either alone, but smaller than the sum of both. An exception is seen in line 8, where the combination of POS tagging, supertagging and PP prediction fails to better the results of just POS tagging and supertagging (line 4). Individual inspection of the results suggests that the lexicalized information of the PP attacher is often counteracted by the less informed predictions of the supertagger (this was confirmed in preliminary experiments by a gain in accuracy when prepositions were exempted from the supertag constraint). Finally, combining all five predictors results in the highest accuracy of all, improving over the first experiment by 2.8% and 3.2% for structural and labelled accuracy respectively.

We see that the introduction of stochastical information into the handwritten language model is generally helpful, although the different predictors contribute different types of information. The POS tagger and PP attacher capture lexicalized regularities which are genuinely new to the grammar: in effect, they refine the language model of the grammar in places that would be tedious to describe through individual rules. In contrast, the more global components tend to make the same predictions as the WCDG itself, only explicitly. This guides the parser so that it tends to check the correct alternative first more often, and has a greater chance of finding the global optimum. This explains why their addition increases parsing accuracy even when their own accuracy is markedly lower than even the baseline (line 1).

6 Related work

The idea of integrating knowledge sources of different origin is not particularly new. It has been successfully used in areas like speech recognition or statistical machine translation where acoustic models or bilingual mappings have to be combined with (monolingual) language models. A similar architecture has been adopted by (Wang and Harper, 2004) who train an n-best supertagger and an attachment predictor on the Penn Treebank and obtain an labelled F-score of 92.4%, thus slightly outperforming the results of (Collins, 1999) who obtained 92.0% on the same sentences, but evaluating on transformed phrase structure trees instead on directly computed dependency relations.

Similar to our approach, the result of (Wang and Harper, 2004) was achieved by integrating the evidence of two (stochastic) components into a single decision procedure on the optimal interpretation. Both, however, have been trained on the very same data set. Combining more than two different knowledge sources into a system for syntactic parsing to our knowledge has never been attempted so far. The possible synergy between different knowledge sources is often assumed but viable alternatives to filtering or selection in a pipelined architecture have not yet been been demonstrated successfully. Therefore, external evidence is either used to restrict the space of possibilities for a subsequent component (Clark and Curran, 2004) or to choose among the alternative results which a traditional rule-based parser usually delivers (Malouf and van Noord, 2004). In contrast to these approaches, our system directly integrates the available evidence into the decision procedure of the rule-based parser by modifying the objective function in a way that helps guiding the parsing process towards the desired interpretation. This seems to be crucial for being able to extend the approach to multiple predictors.

An extensive evaluation of probabilistic dependency parsers has recently been carried out within the framework of the 2006 CoNLL shared task (see `http://nextens.uvt.nl/~conll`). Most successful for many of the 13 different languages has been the system described in (McDonald et al., 2005). This approach is based on a procedure for online large margin learning and considers a huge number of locally available features to predict dependency attachments without being restricted to projective structures. For German it achieves 87.34% labelled and 90.38% unlabelled attachment accuracy. These results are particularly impressive, since due to the strictly local evaluation of attachment hypotheses the runtime complexity of the parser is only $\mathcal{O}(n^2)$.

Although a similar source of text has been used for this evaluation (newspaper), the numbers cannot be directly compared to our results since both the test set and the annotation guidelines differ from those used in our experiments. Moreover, the different methodologies adopted for system development clearly favour a manual grammar development, where more lexical resources are available and because of human involvement a perfect isolation between test and training data can only be guaranteed for the probabilistic components. On the other hand CoNLL restricted itself to the easier attachment task and therefore provided the gold standard POS tag as part of the input data, whereas in our case pure word form sequences are analysed and POS disambiguation is part of the task to be solved. Finally, punctuation has been ignored in the CoNLL evaluation, while we included it in the attachment scores. To compensate for the last two effects we re-evaluated our parser without considering punctuation but providing it with perfect POS tags. Thus, under similar conditions as used for the CoNLL evaluation we achieved a labelled accuracy of 90.4% and an unlabelled one of 91.9%.

Less obvious, though, is a comparison with results which have been obtained for phrase structure trees. Here the state of the art for German is defined by a system which applies treebank transformations to the original NEGRA treebank and extends a Collins-style parser with a suffix analysis (Dubey, 2005). Using the same test set as the one described above, but restricting the maximum sentence length to 40 and providing the correct POS tag, the system achieved a labelled bracket F-score of 76.3%.

7 Conclusions

We have presented an architecture for the fusion of information contributed from a variety of components which are either based on expert knowledge or have been trained on quite different data collections. The results of the experiments show that there is a high degree of synergy between these different contributions, even if they themselves are fairly unreliable. Integrating all the available predictors we were able to improve the overall labelled accuracy on a standard test set for German to 91.1%, a level which is as least as good as the results reported for alternative approaches to parsing German.

The result we obtained also challenges the common perception that rule-based parsers are necessarily inferior to stochastic ones. Supplied with appropriate helper components, the WCDG parser not only reached a surprisingly high level of output quality but in addition appears to be fairly stable against changes in the text type it is applied to (Foth et al., 2005).

We attribute the successful integration of different information sources primarily to the fundamental ability of the WCDG grammar to combine evidence in a soft manner. If unreliable information needs to be integrated, this possibility is certainly an undispensible prerequisite for preventing local errors from accumulating and leading to an unacceptably low degree of reliability for the whole system eventually. By integrating the different predictors into the WCDG parsers's general mechanism for evidence arbitration, we not only avoided the adverse effect of individual error rates multiplying out, but instead were able to even raise the degree of output quality substantially.

From the fact that the combination of all predictor components achieved the best results, even if the individual predictions are fairly unreliable, we can also conclude that diversity in the selection of predictor components is more important than the reliability of their contributions. Among the available predictor components which could be integrated into the parser additionally, the approach of (McDonald et al., 2005) certainly looks most promising. Compared to the shift-reduce parser which has been used as one of the predictor components for our experiments, it seems particularly attractive because it is able to predict non-projective structures without any additional provision, thus avoiding the misfit between our (non-projective) gold standard annotations and the restriction to projective structures that our shift-reduce parser suffers from.

Another interesting goal of future work might be to even consider dynamic predictors, which can change their behaviour according to text type and perhaps even to text structure. This, however, would also require extending and adapting the currently dominating standard scenario of parser evaluation substantially.

References

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: an approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Thorsten Brants, Roland Hendriks, Sabine Kramp, Brigitte Krenn, Cordula Preis, Wojciech Skut, and Hans Uszkoreit. 1997. Das NEGRA-Annotationsschema. Negra project report, Universität des Saarlandes, Computerlinguistik, Saarbrücken, Germany.

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

Thorsten Brants. 2000. TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, WA, USA.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. NAACL-2000*.

Stephen Clark and James R. Curran. 2004. The importance of supertagging for wide-coverage CCG parsing. In *Proc. 20th Int. Conf. on Computational Linguistics, Coling-2004*.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Phd thesis, University of Pennsylvania, Philadephia, PA.

Michael Daum, Kilian Foth, and Wolfgang Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proc. 4th Int. Conf. on Language Resources and Evaluation, LREC-2004*, Lisbon, Portugal.

Amit Dubey. 2005. What to do when lexicalization fails: parsing German with suffix analysis and smoothing. In *Proc. 43rd Annual Meeting of the ACL*, Ann Arbor, MI.

Kilian Foth and Wolfgang Menzel. 2006. The benefit of stochastic PP-attachment to a rule-based parser. In *Proc. 21st Int. Conf. on Computational Linguistics, Coling-ACL-2006*, Sydney.

Kilian A. Foth, Wolfgang Menzel, and Ingo Schröder. 2000. A Transformation-based Parsing Technique with Anytime Properties. In *4th Int. Workshop on Parsing Technologies, IWPT-2000*, pages 89 – 100.

Kilian Foth, Michael Daum, and Wolfgang Menzel. 2005. Parsing unrestricted German text with defeasible constraints. In H. Christiansen, P. R. Skadhauge, and J. Villadsen, editors, *Constraint Solving and Language Processing*, volume 3438 of *Lecture Notes in Artificial Intelligence*, pages 140–157. Springer-Verlag, Berlin.

Kilian Foth, Tomas By, and Wolfgang Menzel. 2006. Guiding a constraint dependency parser with supertags. In *Proc. 21st Int. Conf. on Computational Linguistics, Coling-ACL-2006*, Sydney.

Kilian Foth. 2004. Writing Weighted Constraints for Large Dependency Grammars. In *Proc. Recent Advances in Dependency Grammars, COLING-Workshop 2004*, Geneva, Switzerland.

Jochen Hagenström and Kilian A. Foth. 2002. Tagging for robust parsers. In *Proc. 2nd. Int. Workshop, Robust Methods in Analysis of Natural Language Data, ROMAND-2002*.

Robert Malouf and Gertjan van Noord. 2004. Wide coverage parsing with stochastic attribute value grammars. In *Proc. IJCNLP-04 Workshop Beyond Shallow Analyses - Formalisms and statistical modeling for deep analyses*, Sanya City, China.

Hiroshi Maruyama. 1990. Structural disambiguation with constraint propagation. In *Proc. 28th Annual Meeting of the ACL (ACL-90)*, pages 31–38, Pittsburgh, PA.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proc. Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing, HLT/EMNLP-2005*, Vancouver, B.C.

Joakim Nivre. 2003. An Efficient Algorithm for Projective Dependency Parsing. In *Proc. 4th International Workshop on Parsing Technologies, IWPT-2003*, pages 149–160.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Int. Conf. on New Methods in Language Processing*, Manchester, UK.

Ingo Schröder, Horia F. Pop, Wolfgang Menzel, and Kilian Foth. 2001. Learning grammar weights using genetic algorithms. In *Proceedings Euroconference Recent Advances in Natural Language Processing*, pages 235–239, Tsigov Chark, Bulgaria.

Ingo Schröder. 2002. *Natural Language Parsing with Graded Constraints*. Ph.D. thesis, Dept. of Computer Science, University of Hamburg, Germany.

Martin Volk. 2002. Combining unsupervised and supervised methods for pp attachment disambiguation. In *Proc. of COLING-2002*, Taipeh.

Wen Wang and Mary P. Harper. 2004. A statistical constraint dependency grammar (CDG) parser. In *Proc. ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*, pages 42–49, Barcelona, Spain.