A New Hybrid Dependency Parser for German

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin

Universität Zürich, Institut für Computerlinguistik,
Binzmühlestrasse 14, CH-8050 Zürich

Abstract. We describe the development of a new German parser that uses a hybrid approach, combining a hand-written rule-based grammar with a probabilistic disambiguation system. Our evaluation shows that this parsing approach can compete with state-of-the-art systems both in terms of efficiency and parsing quality. The hybrid approach also allows for the integration of the morphology tool GERTWOL, which leads to a comparatively high precision for core syntactic relations.

1 Introduction

Parsing German keeps attracting interest, both because German is a major European language, and because it has special characteristics such as a relatively free word order and a rich morphology. These characteristics mean that a parsing approach that is appropriate for English is not automatically so for German. While German parsers typically perform worse than English ones, the controversy whether parsing German is an inherently harder task than parsing English is still open [1].

Inter-language comparisons aside, it has been shown that even when only comparing German parsers, choice of treebank and evaluation measure have a considerable effect on reported results [2]. An additional confounding factor is the varying amount of gold information used in different evaluations, ranging from POS-tags up to morphological analyses [3, 4]. We report user-oriented evaluation results that are based on real-world conditions rather than ideal ones. Specifically, only plain text has been taken from the gold set, and all additional information required by the parsers has been predicted automatically.

2 Previous Work on German Dependency Parsing

A number of comparative studies and workshops give estimates of the performance of current German parsers [3–6].

While comparing the results of different studies is not easily possible due to variations in the test setting and evaluation process, conclusions can be drawn from the individual studies.

Kübler, in addition to discussing the particular challenges of German parsing, show encouraging results obtained by statistical constituency parsing (89.18% labeled F-score for a lexicalised Stanford PCFG system) [5]. One needs to bear in

mind, however, that the "constituent structure for a German sentence will often not be sufficient for determining its intended meaning" [5]. This is especially true for noun phrases, which can serve as subjects, different kinds of objects, predicative nouns and genitive attributes, among others. When requiring the correct identification of grammatical function, the parser performs considerably worse (75.33%).

In the PaGe 2008 Shared Task on Parsing German, the dependency version of the MaltParser is shown to be better at identifying grammatical functions than its constituency counterpart and other constituency parsers [6, 7]. The MaltParser is also among the top-performing parsers in both the PaGe 2008 and the CoNLL-X Shared Tasks, obtaining a labeled attachment score of 88.6% and 85.8%, respectively. The labeled attachment score (LAS) measures "the percentage of [non-punctuation] tokens for which the system has predicted the correct head and dependency relation" [4].

Of the 19 participating groups in the CoNLL-X Shared Task on Multilingual Dependency Parsing [4], the average LAS for German is 78.6%. The best parser, described by McDonald et al., achieves 87.3%. Their system "can be formalized as the search for a maximum spanning tree in a directed graph" [8].

Versley compares a parser based on Weighted Constraint Dependency Grammar (WCDG) by [9] to an unlexicalised PCFG parser across different text types, concluding that "statistical parsing for German is lacking not only in comparison with results in English, but also with manually constructed parsers for German" [3]. The better performance of the hand-crafted WCDG parser (an LAS of 88.1% in the TüBa-D/Z test set, in contrast to 79.9% for the PCFG parser) comes at a cost of speed though: parsing took approximately 68 seconds per sentence with the WCDG, and 2 with the PCFG [3].

We respond to the lack of fast German rule-based parsers by presenting a parser that combines a hand-written grammar with a statistical disambiguation system. We report a parsing speed of several sentences per second at 85.5% performance for gold standard POS tags and morphology and 78.4% performance for automatic tagging and morphology.

We use the parser Pro3Gres [10], a fast and robust dependency parser that has been applied widely for English. We have developed a German grammar based on [11].

3 The Pro3Gres Parser

The Pro3Gres parser is a robust and fast bi-lexicalised dependency parser originally developed for English. It uses a hybrid architecture combining a manually written functional dependency grammar (FDG) with statistical lexical disambiguation obtained from the Penn Treebank. The original architecture is shown in figure 1.

The disambiguation method extends the PP-attachment approach of [12] to all major dependency types. The attachment probability for the syntactic relation R at distance $dist$, given the lexical items a and b is calculated using

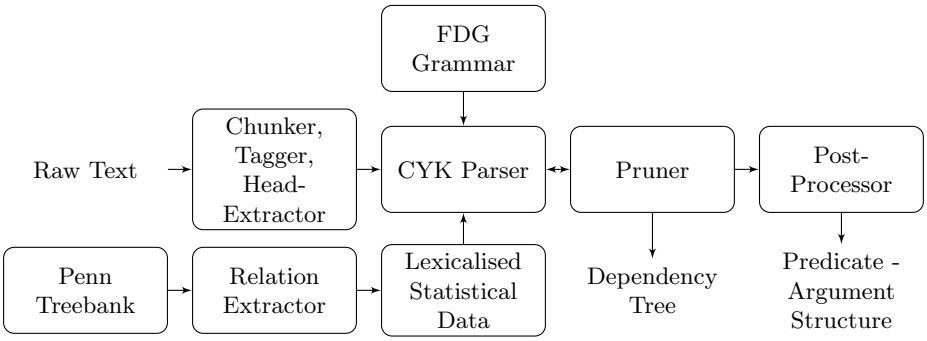

Fig. 1. Pro3Gres architecture

MLE estimation, including several backoff levels.

$$P(R, dist|a, b) \cong p(R|a, b) \cdot p(dist|R) = \frac{f(R, a, b)}{\sum_{i=1}^{n} f(R_i, a, b)} \cdot \frac{f(R, dist)}{fR} \quad (1)$$

The statistical disambiguation allows the parser to prune aggressively while parsing and to return likely analyses that are licensed by the grammar ranked by their probability.

The English Pro3Gres parser has been shown to achieve state-of-the-art performance [10, 13, 14]. We have chosen a modified version of the Pro3Gres system because its architecture has shown to be robust and efficient, making it a promising framework for creating a German parser.

4 Adaptation of Pro3Gres to German

Adapting parsers to new languages and domains has been recognised as an important research area [15]. We have adapted the Pro3Gres parser and its architecture to German in several ways. Taking into account the relatively free word order of German, we have chosen to include morphological and topological rules in the grammar to better identify noun phrase boundaries, in contrast to the English Pro3Gres parser, which uses a dedicated chunker. Existing linguistic resources used include the TreeTagger for POS-tagging, the GERTWOL system for morphological analysis, and part of the TüBa-D/Z corpus for the extraction of statistical data [16–18].

The TüBa-D/Z corpus is a German treebank of written newspaper texts containing approximately 36,000 sentences. We have split the corpus into a training section (32,000 sentences), a development section (1000 sentences), and an evaluation section (3000 sentences).

For the development of the hand-written, rule-based dependency grammar, we used the grammar framework described by [11] as a reference. We used a version of TüBa-D/Z that was automatically converted into this dependency

format, but with some conceptual differences [3]. In case of inconsistency between [11] and TüBa-D/Z, we adopted the labels of the latter. Non-projective structures are processed similarly to pseudo-projective parsing, involving a projective grammar combined with deprojectivization at a later stage [19].

Morphological analysis is an important step in reducing ambiguity, which both improves the speed of the parser and its results. We have used the GERTWOL system to lemmatise all tokens and get their possible morphological analyses. Lemmatising helps to alleviate the sparse data problem in our probabilistic system, and case information marks the grammatical function of a noun phrase. Since many word forms are ambiguous, the lists of possible analyses are of little use in isolation. Only by enforcing agreement rules, both within noun phrases and between subject and verb, can we reduce the degree of ambiguity considerably. Apart from determining grammatical functions, these morphological constraints are also helpful in identifying phrase boundaries.

With about two work-months devoted to the development of the German grammar and the probabilistic disambiguation module, Pro3GresDE works well with common grammatical phenomena, but cannot yet properly handle rarer ones such as genitive objects or noun phrases in vocative or adverbial function.

5 Method for Parser Evaluation

We have already stated that different parsers cannot be directly compared on the basis of their respective performance in various publications. Differences in the test setting, most notably the evaluation measure used and the extent of manual annotation provided, have a considerable effect on the results. Hence, we have decided to conduct a ceteris paribus comparison of Pro3GresDE with two state-of-the-art machine learning parsers, MaltParser [20] and MSTParser [8]. We have evaluated all three parsers using a test set of 3000 sentences of the TüBa-D/Z corpus, with approximately 32,000 sentences being used as a training set.

For every token, our evaluation script tests if the parser predicts the right head and dependency relation. The dependency relation ROOT, which is used for all unattached tokens, including punctuation marks, is ignored in our evaluation. We will report precision, recall and F-scores for total parser performance and selected dependency relations. Additionally, we report the speed of all parsers as measured on an Opteron 8214 system. Since the parsers natively use between one and two of the eight available CPU cores, they have been restricted to using a single core in order to avoid a bias against single-threaded parsers. It is easily possible to run several instances of either parser in parallel to parse large text collections.

In a first round of tests, we measure the performance of Pro3GresDE using different test settings to illustrate the effect of automatically predicting POS-tags (in lieu of gold tags) and morphological information on parser performance. Also, the performance gain achieved by adding statistical disambiguation is shown.

Subsequently, we extend our evaluation to other parsers. All three parsers are provided with the output from TreeTagger, a tokeniser and part-of-speech tagger that was not specifically trained on TüBa-D/Z.[1] It achieves a rather low tagging accuracy of 93.3% on our test set. Tokens tagged with *$.* are considered sentence boundaries, even if this conflicts with the gold set. The sentence/position numbering of the latter has been adjusted to ensure that it is aligned with the parser output for the evaluation. Also included in the test set are lemmas and morphological analyses provided by GERTWOL, although only Pro3GresDE makes use of these. We have not attempted to use this additional data to improve the two machine learning parsers, since lists of morphological analyses are incompatible with the training data of one analysis per token.

6 Evaluation of Pro3GresDE

Table 1 shows total parser performance and speed of Pro3GresDE in different test settings.[2] The F-scores for some dependency relations that are of particular

Table 1. Pro3GresDE precision and recall of the parser over all dependency relations

	binary, nomorph, goldPOS	probabilistic, nomorph, goldPOS	probabilistic, automorph, goldPOS	probabilistic, goldmorph, goldPOS	probabilistic, automorph, autoPOS
precision	68.7	82.9	86.5	88.6	81.5
recall	65.3	79.0	81.2	82.6	75.5
F_1	67.0	80.9	83.8	**85.5**	**78.4**
speed (sentences/sec)	3.3	5.1	7.0	10.9	6.4

interest for tasks such as text mining, are shown in table 2.

Unsurprisingly, the best results are achieved when using the part-of-speech tags and morphological information of the gold standard, the F-score for all relations being 85.5%. While these results are not attainable in a realistic setting with automatic tokenising, POS-tagging and morphological analysis, they come close to other evaluations that used similar test sets [3, 6]. Since Pro3GresDE leaves more tokens unattached than the gold standard, total recall is typically about 6 percentage points lower than total precision.

[1] Instead, the default parameter file was used, which was trained on a newspaper corpus containing 90,000 tokens by IMS Stuttgart.

[2] binary vs. probabilistic: shows whether dependency relations are modified with a (pseudo-)probability or not; nomorph vs. automorph vs. goldmorph: shows whether the test set contains morphological information, and if yes, if it is automatically extracted (GERTWOL) or from the gold standard; goldPOS vs. autoPOS: shows whether the test set uses part-of-speech tags from the gold standard or from an automatic tagger (TreeTagger).

Compared to purely rule-based parsing, we can see that the inclusion of probabilities boosts parser performance by 10 percentage points, while at the same time speeding it up by 50%. While the effect on total performance of adding

Table 2. Pro3GresDE: F_1 for selected dependency relations. (SUBJ: subject; OBJA: accusative object; OBJD: dative object; PRED: predicate; APP: apposition; PP: prepositional phrase as adjunct; OBJP: PP as complement)

	binary, nomorph, goldPOS	probabilistic, nomorph, goldPOS	probabilistic, automorph, goldPOS	probabilistic, goldmorph, goldPOS	probabilistic, automorph, autoPOS
SUBJ	51.9	83.3	89.1	93.0	82.5
OBJA	25.2	**64.6**	**81.2**	90.0	74.2
OBJD	8.4	**29.2**	**63.8**	81.8	56.3
PRED	**19.0**	**67.2**	67.4	69.6	58.8
GMOD	50.0	**62.6**	**86.2**	94.2	81.0
APP	40.0	72.9	76.3	79.5	66.9
PP	53.8	70.0	69.6	70.4	64.0
OBJP	**20.7**	**70.2**	69.4	70.0	61.3

automatically extracted morphological information is not as big – still a considerable improvement of 3 percentage points – case information is very helpful in attributing the correct function to noun phrases, increasing the F-score for accusative objects by 16, for dative objects by 34, and for genitive modifiers by 24 percentage points. Additionally, it boosts the speed by further 40%. While it might seem counterintuitive that parsing speed increases as the system becomes more complex, the additional modules allow us to discard unlikely or morphologically unsound analyses at an early stage, which reduces the number of ambiguous structures that have to be built up.

The parser is optimised for best results with both the statistics module enabled and GERTWOL morphology information available. If the statistics module is disabled, we have observed that increasing the complexity of the rule-based grammar resulted in a decrease in parser performance. This is due to rare dependency relations such as PRED (predicate) and OBJP (prepositional phrase as verb complement) being heavily overpredicted in this test setting. In both cases, the label is distinguished from structurally identical ones on a semantic level [11], and both relations only occur with certain verbs. Using a probabilistic disambiguation, we can improve the F-score for PRED from 19.0% to 67.2%, and for OBJP from 20.7% to 70.2%.

Similarly, little time has been invested into improving parser performance when morphological information is missing. Hence, the parser will even consider morphologically unambiguous pronouns such as *ihn* to be possible subjects, a problem which could be solved by using full bilexicalization in the absence of morphological information.

The dependency relation labeled APP, which covers proper appositions, is also used to link tokens within multi-noun chunks. Consequently, the APP results in table 2 are an indication of how well NP boundaries are recognised by the parser. Probabilistic rules result in a 33 percentage point increase in performance. Morphological information leads to a further improvement (3 and 6 points for automatic and gold morphological information, respectively). APP is one of the relations with a higher recall than precision (82.3% versus 76.8% with gold morphology and tagging), which indicates that too few phrase boundaries are predicted.

Using automatic POS-tagging, parsing results are considerably worse, with an F-score of 78.4% for all relations. The performance drop of 5.4 percentage points is close to the error rate of the POS-tagger (6%), but we deem this to be coincidental. This is because tagging errors are of varying significance. Whereas the distinction between proper names and nouns is of relatively little importance in our grammar, erroneously tagged verbs may lead to all verbal dependents being incorrectly attached.

7 Comparing Pro3GresDE to MaltParser and MSTParser

While Pro3GresDE does not quite reach the performance that has been reported for other parsers, a comparison based on different corpora, evaluation scripts etc. is of little relevance. When parsing the test set that has been described above

Table 3. Parser performance (total results and F_1 scores of selected grammatical relations)

	Pro3GresDE	MaltParser	MSTParser
Precision	81.5	79.1	78.9
Recall	75.5	79.5	76.5
F_1	78.4	79.3	77.7
SUBJ	82.5	77.1	75.5
OBJA	74.2	64.8	65.8
OBJD	56.3	29.4	31.4
GMOD	81.0	69.6	71.5
ADV	66.8	76.2	79.0

with MaltParser and MSTParser, the two parsers obtained considerably lower scores than in CoNLL-X. This performance drop was to be expected due to differences in the test setting. Most importantly, parser input has more noise, with automatically assigned POS-tags and sentence boundaries instead of the gold ones.

Another factor that might explain the relatively low performance of MaltParser and MSTParser is that both parsers, albeit trained on TüBa-D/Z, were

not specifically tuned for best results on this corpus. For MaltParser, we used CoNLL-X settings, including pseudo-projective parsing [19]. The latter led to a 10 percentage point increase in recall, albeit with a 5 percentage point loss in precision. We have used standard settings for MSTParser, with the only exception that we chose the non-projective parsing algorithm, which in preliminary tests outperformed the projective one.

The fact that Pro3GresDE was developed on TüBa-D/Z puts it at an advantage in this evaluation. It is unclear how evaluation results would be affected by more neutral test data, with the training data staying the same. Versley, when comparing a rule-based WCDG parser and a statistical PCFG one, found that both were "equally sensitive to text type variation" [3]. When using another treebank for both training and testing, we expect the statistical parsers to have the advantage. They can be retrained on a portion of the same treebank they are evaluated on, while rule-based parsers require a (often lossy) mapping between the different dependency representations [13].

Regarding general performance, we can observe that the total F-scores of MaltParser, MSTParser and Pro3GresDE seem very similar, with the gap between the best-performing and the worst-performing system being 1.6 percentage points. On closer inspection, however, the results of the parsers are clearly different. The variance between parsers is greater when considering precision and recall instead of the F-score. Pro3GresDE achieves the highest precision, but the lowest recall, while MaltParser features a recall that is slightly higher than its precision. A recall lower than precision indicates that the parser predicted more unattached tokens (root nodes) than exist in the gold set, and vice versa.

Parser performance also varies when analysing single dependency relations. From this point of view, Pro3GresDE has some clear strengths and weaknesses. For the dependency relation ADV, the performance of Pro3GresDE is more than 10 percentage points worse than that of MSTParser (66.8% and 79.0% F-score respectively). This is mainly due to the fact that the Pro3GresDE grammar attaches adverbs to the finite verb if possible, without considering all possible heads. This leads to a high number of tokens that are correctly identified as adverbs, but attached to the wrong head (which accounts for 60-70% of the errors).[3]

On the other hand, Pro3GresDE performs better than the machine learning systems when it comes to the grammatical function of noun phrases. For the dependency relations SUBJ, OBJA, OBJD and GMOD, Pro3GresDE outperforms the other parsers by 5 to 25 percentage points. This is possible through the inclusion of automatically extracted morphological information. The high ambiguity of a morphological analysis, which is only resolved in the parsing process itself through agreement rules, makes it unlikely that machine learning systems can successfully integrate this data to improve parsing performance. Hence, we consider the ability to use highly ambiguous morphological information to increase parser performance an advantage of our rule-based system.

[3] A full disambiguation of adverb attachment can be computationally expensive with our approach. So far, we chose to focus on other syntactic relations.

MSTParser was the fastest parser in our evaluation, parsing our test set 40% faster than MaltParser and 30% faster than Pro3GresDE. These differences are small, however, compared to the factor 30 speed difference reported in [3]. Parser settings and the training set are likely to have a bigger effect on parsing speed than parser choice.[4] Still, all three parsers reach a parsing speed of several sentences per second and are thus applicable for large-scale parsing tasks.

Table 4. Parse time (for 3000 sentences) and speed (in sentences per second)

Parser	Time	Speed
Pro3GresDE	467s	6.4
MaltParser	604s	5.0
MSTParser	438s	6.8

8 Conclusions

In summary, Pro3GresDE achieves competitive results, both in terms of efficiency and performance. Used in combination with GERTWOL, it outperforms MaltParser and MSTParser in the prediction of central grammatical relations such as subjects and objects, a property which makes the parser a suitable choice for tasks relying on this information.

Future research will include an extension of statistical disambiguation rules and integration of additional linguistic resources to further improve parser performance.

References

1. Kübler, S.: How do treebank annotation schemes influence parsing results? Or how not to compare apples and oranges. In Nicolov, N., Boncheva, K., Angelova, G., Mitkov, R., eds.: Recent Advances in Natural Language Processing IV: Selected Papers from RANLP 2005, Amsterdam, John Benjamins (2006)
2. Rehbein, I., van Genabith, J.: Why is it so difficult to compare treebanks? TIGER and TüBa-D/Z revisited. In: Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories, Bergen, Norway (2008)
3. Versley, Y.: Parser evaluation across text types. In: Fourth Workshop on Treebanks and Linguistic Theories (TLT), Barcelona, Spain (2005)

[4] Specifically, parsing time of MaltParser depends on the number of support vectors, which grows with training set size. For MSTParser, parsing time is independent of training set size, but depends on the number of features used (Joakim Nivre, personal communication, March 24, 2009).

4. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), New York City, Association for Computational Linguistics (June 2006) 149–164
5. Kübler, S., Hinrichs, E.W., Maier, W.: Is it really that difficult to parse German? In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP), Sydney, Australia (2006)
6. Kübler, S.: The PaGe 2008 shared task on parsing German. In: Proceedings of the Workshop on Parsing German, Columbus, Ohio, Association for Computational Linguistics (June 2008) 55–63
7. Hall, J., Nivre, J.: A dependency-driven parser for German dependency and constituency representations. In: Proceedings of the ACL 2008 Workshop on Parsing German, Columbus, Ohio (2008) 47–54
8. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: Proceedings of HLT-EMNLP. (2005)
9. Foth, K.A., Daum, M., Menzel, W.: A broad-coverage parser for German based on defeasible constraints. In: KONVENS 2004, Beiträge zur 7. Konferenz zur Verarbeitung natürlicher Sprache, Vienna, Austria (2004)
10. Schneider, G.: Hybrid Long-Distance Functional Dependency Parsing. Doctoral Thesis, Institute of Computational Linguistics, University of Zurich (2008)
11. Foth, K.A.: Eine umfassende Contraint-Dependenz-Grammatik des Deutschen. University of Hamburg (2005)
12. Collins, M., Brooks, J.: Prepositional attachment through a backed-off model. In: Proceedings of the Third Workshop on Very Large Corpora, Cambridge, MA (1995)
13. Schneider, G., Kaljurand, K., Rinaldi, F., Kuhn, T.: Pro3Gres parser in the CoNLL domain adaptation shared task. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, Prague (2007) 1161–1165
14. Haverinen, K., Ginter, F., Pyysalo, S., Salakoski, T.: Accurate conversion of dependency parses: targeting the Stanford scheme. In: Proceedings of Third International Symposium on Semantic Mining in Biomedicine (SMBM 2008), Turku, Finland (2008)
15. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007. (2007) 915–932
16. Schmid, H.: Probabilistic part–of–speech tagging using decision trees. In: Proceedings of the Conference on New Methods in Language Processing, Manchester, UK (1994)
17. Haapalainen, M., Majorin, A.: GERTWOL und Morphologische Disambiguierung für das Deutsche. In: Proceedings of the 10th Nordic Conference of Computational Linguistics, University of Helsinki, Department of General Linguistics (1995)
18. Telljohann, H., Hinrichs, E.W., Kübler, S.: The TüBa-D/Z treebank: Annotating German with a context-free backbone. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation, Lisbon, Portugal (2004)
19. Nivre, J., Nilsson, J.: Pseudo-projective dependency parsing. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, Michigan, Association for Computational Linguistics (June 2005) 99–106
20. Nivre, J., Hall, J., Nilsson, J.: Maltparser: A data-driven parser-generator for dependency parsing. In: Proceedings of LREC, Genoa, Italy (2006) 2216 –2219