Because Size Does Matter: The Hamburg Dependency Treebank

Kilian Foth, Arne Köhn, Niels Beuck, Wolfgang Menzel
Fachbereich Informatik
Universität Hamburg
{foth, koehn, beuck, menzel}@informatik.uni-hamburg.de

Abstract

We present the Hamburg Dependency Treebank (HDT), which to our knowledge is the largest dependency treebank currently available. It consists of genuine dependency annotations, i. e. they have not been transformed from phrase structures. We explore characteristics of the treebank and compare it against others. To exemplify the benefit of large dependency treebanks, we evaluate different parsers on the HDT. In addition, a set of tools will be described which help working with and searching in the treebank.

Keywords: Dependency Treebank, German, Parser Evaluation

1. Introduction

In recent years, dependencies have become increasingly popular for encoding syntactic structure: The CoNLL shared tasks from 2006 to 2009 (Buchholz and Marsi, 2006; Nivre et al., 2007; Surdeanu et al., 2008; Hajič et al., 2009) all used dependency annotations. Also, there is a wide variety of high quality dependency parsers based on different machine learning paradigms and decision procedures (e. g. Nivre (2003), McDonald et al. (2005), Martins et al. (2009), Huang and Sagae (2010), Bohnet (2010)).

Treebank construction, however, has not yet caught up with this trend. For several languages such as German (TIGER (Brants et al., 2004), TüBa-D/Z (Telljohann et al., 2004)) and English (PTB (Marcus et al., 1994)), the primarily used dependency treebanks are automatically generated from phrase structure annotations. In this process, heuristic transformations need to be applied, and therefore the resulting dependency annotations are not as reliable.

We have created a genuine dependency treebank for German which is --- as far as we know --- the largest dependency treebank available with nearly four million hand-annotated tokens. It contains three to four times as many manually annotated tokens as the TIGER treebank, the Penn Treebank, the Chinese Treebank and the Spoken Dutch corpus. It is also more than twice as large as the Prague Dependency Treebank and TüBa-D/Z. Contrary to other reports (Ballesteros et al., 2012), we will show that such a huge corpus actually does pay off in terms off better parsing accuracy.

The Hamburg Dependency Treebank consists of 261,830 German sentences annotated with dependency structures, which have been encoded using different degrees of manual effort. The treebank contains 101,999 sentences with high-quality annotation, which have been produced by manual revision and a subsequent cross-checking for consistency. Further 104,897 sentences have been manually revised but not checked for consistency. The remaining 54,934 sentences are annotated with raw parser output. The whole corpus is available free of charge for scientific purposes[1].

This paper is structured as follows: Section 2 describes the annotation process, Section 3 reports the consistency checks carried out and the resulting changes, and Section 4 presents some characteristics of the resulting treebank. Section 5 gives an overview of the software that is shipped with the data, Section 6 reports parser evaluations on the HDT and Section 7 concludes.

2. The Data Source and its Annotation

The raw text of the treebank is formed by online newscasts of the technical news service `www.heise.de`; all news items were taken from the years 1996--2001. This source was chosen for being freely redistributable, for being available in large and steadily growing quantity, and for covering a domain which is only partially restricted. The content of the articles ranges from formulaic periodic updates on new BIOS revisions and processor models or quarterly earnings of tech companies over features about general trends in the hardware and software market to general coverage of social, legal and political issues in cyberspace, sometimes in the form of extensive weekly editorial comments. The mapping from sentences to articles and authors is retained, allowing, e. g. analysis of individual style. The creation of the treebank through manual annotation was largely interleaved with the creation of a standard for morphologically and syntactically annotating sentences as well as a constraint-based parser.

The original Stuttgart-Tübingen Tag Set (STTS) for German (Schiller et al., 1999) was used for morphological classification of words, and a pre-existing dependency model of German with limited coverage was chosen as the starting point for the target of syntactical annotation. Over the course of annotation, this model was expanded to wide coverage of unrestricted German input; its final form provides 35 different subordination labels to distinguish syntactic functions such as direct and oblique objects, obligatory and optional subordination between open-class and function words of all classes in the STTS. The annotation guidelines are described in detail in the annotator's manual (Foth, 2006a). In addition to the syntactic dependencies, an extra-syntactic reference specifies the antecedent of relative pronouns independent of their function in the subclause.

2.1. The Annotation Process

An existing constraint dependency analyzer (Schröder, 2002) was used to create an approximate analysis for the unannotated sentences. Although these initial analyses

[1] `http://nats-www.informatik.uni-hamburg.de/HDT/`

were often far from the desired result, they nevertheless provided a more efficient starting point for manual annotation than any attempt to construct each tree from scratch would have been. As the model was defined and the rules of the constraint dependency grammar were refined to deal with more phenomena and to resolve more ambiguities reliably, unannotated portions of the corpus were periodically re-parsed with the current constraint dependency grammar to improve the quality of the suggested dependency trees.

Annotation was performed using a graphical tool (Foth et al., 2004) which uses the same constraint evaluation engine as the parser. Even more, the same defeasible constraints that guide the transformation-based parsing algorithm (Foth et al., 2000) are used to provide visual feedback to the annotators: each morphological variant, dependency label and dependency edge is displayed in green or red hues depending on the strictness of the violated constraints. The tool also displays a sentence-wide penalty score which is computed from the violated constraints. Interactive re-attachment and re-labelling automatically updates this information.

An analysis that violates constraints can take one of three forms. It is always possible that the parser failed to find the syntactically most appropriate analysis due to the heuristic nature of the solution method (search error). Such erroneous suggestions are simply edited by the annotator to conform to the grammar as it is. In this case, the penalty score improves, which indicates that the modification is appropriate. In other cases a suggestion is actually rated higher by the weighted constraints than the version preferred by humans, i. e. the verified tree is non-optimal according to the current parsing grammar (model error). This points to an opportunity to refine the grammar. The third possibility is that an utterance genuinely violates a preference that is shared by human and machine, i. e. exhibits dispreferred behaviour that is justified by higher-level (e. g. suprasentential) factors; as expected, such ‘marked’ phenomena are rarer than the first two forms.

The direct demonstration of mistaken human assumptions provided by the second form was a major driver of grammar development. The proliferation of new constraints and extensions or exceptions to existing ones had to balance a wider coverage with maintaining correct grammaticality judgements on existing phrase and sentence types. Because of the high cost of calculating optimal analyses for a big corpus, it is usually not possible to prove formally that covering a new phenomenon does not decrease accuracy when analyzing an old one. However, a full record of the constraints violated by every verified tree was kept as a countermeasure; if a change in the grammar causes the preferred analysis of a previous sentence to violate additional serious constraints, this points to an oversight in the proposed change so that it has to be revised or renounced altogether. Periodically, portions of the corpus were analysed from scratch and compared to the verified analyses to check that developing the grammar further did not decrease the overall parsing accuracy.

In this way, automatic analysis was gradually improved so that the grammaticality judgments of the current grammar can help annotators with semi-automatic correction of errors committed during automatic analysis by an earlier version.

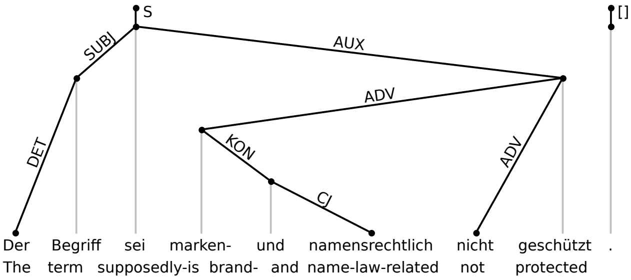
S
[]
SUBJ
AUX
ADV
DET
KON
ADV
CJ
Der Begriff sei marken- und namensrechtlich nicht geschützt .
The term supposedly-is brand- and name-law-related not protected

Figure 1: Graphical representation of a sample annotation

2.2. The Annotation Scheme

The dependency model of German used for this treebank was constructed to provide a robust coverage of all phenomena that can be expected to occur repeatedly in normal written text, while adequately representing the richness of the occurring relations. Thus, it not only represents the subordination relation between two words in its structure but also indicates the many different types of subordination via labels. For instance, we distinguish not only complements from adjuncts but also subjects from objects and several types of direct, indirect and prepositional objects, as well as SVO and SOV object clauses.

At the same time, a level of discriminatory power was chosen that reflects the limit of the disambiguating decisions a syntax-based dependency parser can reasonably make. An example of a decision that is too subtle to be made reliably would be the distinction between defining and non-defining relative clauses. Although there can be a large difference in meaning between two relative clauses with the same surface reading, resolving this ambiguity usually requires a large amount of knowledge about quantors, relations and entities in the real word that is far beyond the capabilities of the most sophisticated word-to-word subordination model. Therefore, we indifferently annotate both kinds of relative relations between sentences as ‘REL’.

Referential relations are dealt with on a separate level, i. e. a second dependency structure over the same words. Currently, the relation between a relative pronoun and its antecedent is the only referential relation that is annotated. Other references are often too ambiguous to pin down and in any case usually transcend the powers of a *sentence* as opposed to a *text* analyzer, because they would have to connect words across sentence boundaries.

The dependency labels used for annotation are described in detail in (Foth, 2006a), albeit in German. Therefore, we give a brief overview in English. For every label the relative number of occurences in the manually annotated part of the treebank is given.

ADV Denotes adverbial modification by proper adverbs or words from related classes (predicative adjectives and various particles that the STTS assigns to their own class) *7.026%*

APP (apposition, always subordinated strictly left to right) Relates adjacent nominal words in the same NP (headline phrases) or in proper appositions (I, Robot)

4.172%

ATTR Attributive adjectives or numbers modifying a noun *7.301%*

AUX Auxiliary, connects verbs in the same verb group, the finite verb is always the head of such a chain *3.396%*

AVZ (Abtrennbarer VerbZusatz) separable verb particle, attaches a separated verb particle to its verb *0.587%*

CJ Conjunct, complement of a conjunction, i. e. connected to a word like 'und' *2.828%*

DET Determiner of a noun *12.251%*

ETH Ethic dative, i. e. a nominal adjunct in the dative case that is not licensed by a verb frame *0.073%*

EXPL (expletive) only used for the expletive use of the pronoun 'es' *0.09%*

GMOD Genitive modification, the dependent word is in the genitive case and modifies a nominal *2.202%*

GRAD Gradual, an NP indicating a measurement as in "three meters deep" *0.056%*

KOM Comparison words modifying a noun or a verb, typically 'wie' or 'als' *0.588%*

KON Coordination connecting words in a coordination chain (except the final word below a coordination, which is CJ). In coordinations, the word to the left is always the head of the word to the right *2.903%*

KONJ Conjunction modifying a verb signalling an SOV subclause *0.873%*

NEB (Nebensatz) Subordinate clause, connecting the finite verb of the subordinate clause to the verb in the superordinate main clause. (For some types of subclauses, such as relative clauses, there are special labels.) *0.66%*

NP2 A rare label for logical subjects in elliptical coordinations *0.02%*

OBJA Accusative object *4.013%*

OBJA2 Second accusative object, for the rare case where a verb has a valency for two accusative objects *0.049%*

OBJC Object clause, for the finite verb in a subclause that is attached to a verb as a complement *0.247%*

OBJD Dative object *0.406%*

OBJG Genitive object *0.016%*

OBJI Infinitive verb used as a complement to another verb *0.379%*

OBJP Prepositional object, for prepositions that are a complement to a verb. In contrast to a PP, it cannot be omitted. *0.442%*

PAR Parenthesis, superior clause that is inserted into its subclause. In such a case, to prevent a non-projective structure, the finite verb of the subclause is attached to the last word before the inserted clause. *0.042%*

PART Particle, for example 'zu' modifying an infinite verb, or the second part of a circumposition modifying the respective preposition *0.528%*

PN The complement of a preposition (or post-position) *10.726%*

PP Prepositional phrase, for the attachment of prepositions *10.587%*

PRED Predicative complement, mostly for the verb 'sein' *0.998%*

REL (relative clause) Connects the finite verb of a relative clause to its (nominal or verbal) antecedent. Often non-projective. *0.837%*

S (sentence) the label for the root node of SVO sentences and phrase fragments, or an SVO sentence subordinated to a verb as a complement. *6.001%*

SUBJ (surface subject) Any nominal material filling the subject slot of a verb (not necessarily the vorfeld position, see 'EXPL') *7.250%*

SUBJC (subject clause) Any verbal material filling a subject slot *0.182%*

VOK (Vokativ) Salutation, usually a proper name, arbitrarily attached to the nearest word because of its tenuous connection with the syntax tree *0.002%*

ZEIT (time) Time information in the form of (usually four-digit) year numbers attached without a preposition *0.34%*

" (the empty label) for punctuation marks *11.93%*

REF The only label for the separate reference level: the label of pronouns attached to their antecedent.

In contrast to the set of 34 dependency labels, which was refined over time and which could arguably have turned out somewhat smaller or larger, we consider the decisions about word-to-word subordinations largely unproblematic. For the most contested issues in dependency subordination, we simply chose one position and adhered to it consistently. For instance, our determiners are attached below the noun they accompany; multi-part verb phrases are always headed by the finite verb; and verb complements are always attached to the full verb rather than to an auxiliary verb. Neither of these decisions should be viewed as a linguistic statement, e. g. about the reality of determiner phrases as opposed to noun phrases; if an NLP system requires determiners to be superordinated, it would be easy to exchange the direction of all 'DET' dependencies systematically.

3. Quality Assurance

To assure the consistency of the annotation, we applied the DECCA tools (Boyd et al., 2008) to a substantial part of the corpus, which check the annotation in two independent steps, one for the part-of-speech tags and the other for dependency labels. In both cases the approach is similar: an algorithm identifies where similar structures are annotated differently. These hints are then inspected manually to decide where changes to the annotations are necessary.

For the dependency labels, the algorithm determines for every pair of head and dependent in an annotation, whether the same two words are connected in other sentences via a different label. The automatic consistency check pointed out 8495 such word pairs. Manual investigation found out that for 1931 of them at least one occurrence was indeed erroneous and therefore had to be changed. The resulting precision of the automatic consistency check, based on word pairs, is 22.7%. The recall can naturally not be determined this way, as only those annotations pointed out by the tool where examined again.

The top six changes are given in Table 1. Note that in some cases more than one change per entry was necessary. Therefore, the numbers of individual changes add up to more than 1931. The most common change (1021 cases) was replacing the prepositional phrase (PP) label with the prepositional object (OBJP) label, i. e. switching from an adjunct

from	to	#changes
PP	OBJP	1021
ADV	AVZ	421
APP	ZEIT	347
SUBJ	OBJA	300
OBJA	SUBJ	298
ADV	PRED	291

Table 1: Most applied corrections from cross-checking with DECCA

to a complement reading. The reverse case (OBJP ⇒ PP) is much less common (75 cases). The large number of PP ⇒ OBJP changes might be explained by the fact that this distinction is somewhat ambiguous and final ruling of the annotation guideline was the result of an iterative process. Sentences annotated at the beginning of this process where not always revised with respect to the final guideline.

The inconsistency between the dependency types ‘ADV’ and ‘AVZ’ is almost entirely due to German *Funktionsverbgefüge*, which derive from free adverb adjuncts with the same surface form, and which are usually indistinguishable in meaning as well as in form from the adjunct reading. The label ‘ZEIT’, used for asyndetic combination of year numbers to other noun and verb phrases, was added to the annotation model relatively late, and had often originally been labeled ‘APP’ merely because there was no other possible noun-noun subordination. Even so, there is a residue of uncertainty about such combinations; ‘Olympia 2004’ is certainly an instance of an attributive year number, but ‘Windows 95’ is less clear-cut, and ‘Ipse 2000’ even less so. The confusion between subjects and objects is much more serious; it arose mostly in morphologically ambiguous transitive main clauses (note the almost identical incidence numbers). The automatic parser, oblivious to meaning, consistently prefers the SVO solution to avoid the (small) ‘Inversion’ penalty in the grammar; even where this was obviously wrong, confirmation bias often induced human reviewers to overlook such errors. The confusion pair ‘ADV’ vs. ‘PRED’ expresses the difference between a predicative adverb and a merely adjunctive one; here, again, the distinction is often hard to discern, and must consistently be established in an arbitrary way.

While 0.5 sentences were changed per word pair on average, the word pair ‘bekannt gegeben’ alone resulted in 109 changes (ADV ⇒ AVZ). All in all, the consistency check for dependency labels resulted in adjustments of 4% of the sentences.

4. Treebank characteristics

Sentences in the HDT have an average length of 18.4 tokens. The longest sentence consists of 144 tokens. The number of different word forms is quite high: there are 130,933 different word forms (this number shrinks to 126,801 when ignoring case). 77,397 of them appear only once. This is due to the large amount of technology-related compounds that are mentioned in the corpus such as “3,5-ZOLL-Wechselplatte” (3.5 inch removable hard disk drive).

Property	HDT	CoNLL-X German
non-projective	12.52%	27.75%
non-planar	10.89%	27.71%
ill-nested	0.51%	1.06%

Table 2: Percentage of dependency trees violating projectivity, planarity, and well-nestedness. Only the manually annotated parts are considered for HDT.

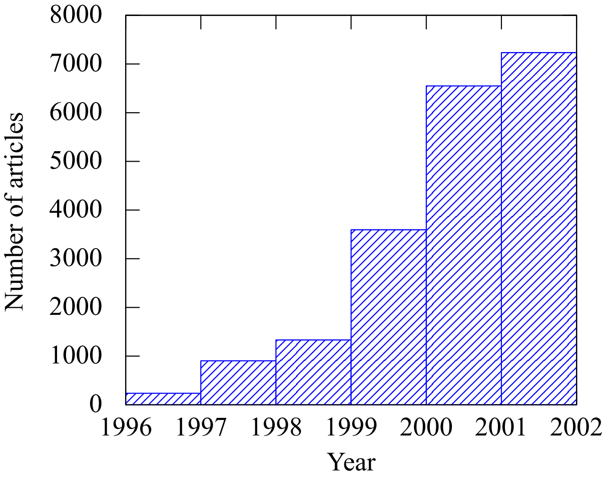

Figure 2: Distribution of the articles by year

The HDT contains non-projective, non-planar as well as ill-nested sentences. The fractions of sentences violating these properties are shown in Table 2. Havelka (2007) discusses these properties in detail and provides an evaluation for all treebanks contained in the CoNLL-X shared task.

The HDT contains only about half as many sentences violating each property. This could be due to the fact that the HDT annotation guidelines try to avoid nonprojective structures. Nonetheless, the HDT still has a higher percentage of ill-nested sentences than the CoNLL-X datasets other than German. The label distribution for non-projective arcs is given in Table 3.

The articles in the HDT were written between 1996 and 2001. Figure 2 shows the distribution by year.

We have studied how much the different parts of the annotation correlate and where a parser is faced with particularly easy or hard labelling decisions. For a single edge, the pair of PoS tag of the head and the dependent is a very good predictor of the label: While the PoS tags uniquely determine the label only in about 20% of the instances, a maximum likelihood guesser would already achieve an accuracy of nearly 91%.[2] There are only very few combinations where the dependent’s PoS uniquely predicts the head’s PoS. Here, a maximum likelihood guesser would achieve an accuracy of about 49%.

5. Accompanying software

To work with the corpus, we also provide a toolbox containing parsers for the HDT file format, which are written in

[2] Punctuation marks are always trivially attached to NIL with the empty label and have therefore been excluded for this estimate.

Label	PP	OBJA	ADV	REL	KON	OBJD	APP	other
Non-projective arcs	23.15%	19.44%	17.99%	10.85%	6.43%	3.87%	3.70%	14.57%

Table 3: Label distribution for non-projective arcs of the manually annotated part of the HDT

python and Java, a tool to convert HDT files into the widely used CoNLL-X format, the scripts that have been used for gathering the statistics in Section 4 and other little helpers, e. g. for stripping annotations and generating sentence prefixes.

There is also a web interface -- cobacose[3] -- for searching the corpus by means of constraints. It uses the same constraints as WCDG does for parsing. This provides a powerful query language specifically tailored to dependency structures.

6. Parser Evaluation on the HDT

To train a parser optimally, one needs a large treebank of high quality. With the availability of the Hamburg Dependency Treebank, it becomes possible to estimate the gain that can be expected from additional data of varying quality.

In this section, multiple parsers will be evaluated on different subsets of the HDT. This way, several aspects can be studied:

- The impact of the size of the training set on parsing accuracy
- the influence of data quality on parsing results, and
- the benefit of adding lower quality data to a high-quality training set.

6.1. The Parsers

We used three different parsers in our evaluation: MaltParser, the Bohnet parser, and TurboParser. These were selected because they represent different approaches to parsing, are able to create non-projective structures and are freely available.

MaltParser is a transition-based parser that can produce non-projective dependency trees when using the 2-planar algorithm (Gómez-Rodríguez and Nivre, 2010). It employs a greedy strategy by picking the locally best action at every parsing step. We use MaltParser version 1.7.2 in the default 2-planar configuration.

The Bohnet parser (Bohnet, 2010) is a graph-based dependency parser that uses the second order maximum spanning tree algorithm of Carreras (2007) and the non-projective approximation algorithm described in McDonald and Pereira (2006).

TurboParser (Martins et al., 2009) is another graph-based dependency parser that uses features similar to the parser described in Carreras (2007). Because the problem of finding the optimal tree is intractable when allowing non-projective solutions, an approximating algorithm is used instead: The task is converted to an integer linear programming problem, which is then solved approximately. This way, non-projective parses can be generated directly in contrast to the approach taken by the Bohnet parser. The version of TurboParser evaluated in this work is the one described in Martins et al. (2013).

6.2. Experimental Setup and Results

As previously noted, the Hamburg Dependency Treebank consists of three parts, that are annotated with different degrees of revision effort. These parts are:

A) automatically parsed, manually corrected and cross-checked for consistency (101,999 sentences)

B) automatically parsed and manually corrected (104,897 sentences)

C) automatically parsed without revision (54,934 sentences)

Every parser has been trained on a 10, 100, 1000, 10,000, 50,000 and 100,000 sentence subset of both the parts A and B. Furthermore, the parsers have been trained on subsets of up to 50,000 sentences of part C. The sentences 100,001 to 101,999 of part A have been used for evaluation in every experiment.

The results for the labeled attachment score over training data size can be seen in Figure 3 and Figure 4. TurboParser and the Bohnet parser perform best while MaltParser ranks third. The good result of the Bohnet parser relative to TurboParser is particularly noteworthy since a comparison of the results reported in Martins et al. (2013) and Bohnet (2010) suggests a remarkable difference.

As can be seen in Figure 3, the more data is used for training the better the accuracy becomes. Given enough data, both the Bohnet parser and TurboParser achieve higher accuracies on the HDT than the highest ones reported for other treebanks so far: TurboParser's highest unlabeled attachment score on the CoNLL-X data set (Buchholz and Marsi, 2006) was 93.52% (for Japanese) (Martins et al., 2013) and the highest labeled attachment score reported by Bohnet (2010) is 90.33% on the English CoNLL-2009 data set (Hajič et al., 2009).

To test whether automatically annotated data can help the parser, the 1,000 sentences subset of part A has been mixed with different subsets of Part C. The results show that adding sentences out of part C of the treebank yields better parsing results (see Figure 4), or the other way round: adding a certain amount of high quality data increases the value of a data set of lower quality. However, adding low quality data to a fairly big amount of high quality data can even worsen the accuracy: When the 50,000 sentence subset of part A is used for training in conjunction with the 50,000 sentence subset of part C, the accuracy for TurboParser drops from 93.57% to 92.84%, the same happens for the Bohnet parser (from 93.93% to 92.61%) and to a lesser extent for MaltParser (from 85.56 to 85.00%).

[3] The *co*nstraint-*ba*sed *co*rpus *se*arch

training size	10	100	1,000	10,000	50,000	100,000
unknown word forms	8817	8352	6749	3828	2375	1624

Table 4: Unknown word forms in the evaluation set w.r.t. the subsets of part A

The accuracies achieved by training the parsers on data from A and B only differ slightly, whereas training on C (i. e., automatically parsed data) leads to a significantly worse accuracy. However, it is noteworthy that even in this case both the Bohnet parser and TurboParser reach an accuracy that is close to the one of WCDG as reported by Foth (2006b) (90.9%). Note that this is the parser used to generate the annotation for part C of the treebank.

TurboParser and the Bohnet parser benefit significantly more from a larger training set than MaltParser: The first two yield an error reduction of 14% and 13%, respectively, when increasing the amount of sentences in the training set from 50,000 to 100,000 sentences, whereas MaltParser only achieves an error reduction of 5%. This shows that the parsing approaches do not just differ in their general parsing accuracy but also in their ability to profit from more training data.

Bohnet (2010) argues that one of the benefits of the Hash Kernel used in his parser is that the "Hash Kernel provides an over-proportional accuracy gain with less training data compared to MIRA". However, our results do not support this claim as they show that the Bohnet parser actually has a slightly worse accuracy than TurboParser (which uses MIRA) when given only small amounts of training data. However, the data does not allow for a general comparison between these two approaches because the two parsers are based on considerably different principles.

7. Conclusion

We presented the Hamburg Dependency Treebank, a large-scale corpus of German newscast complemented by a fairly rich annotation, which combines syntactic relationships between words with an additional reference specification for relative pronouns and a detailed morphological characterization of the tokens.

The treebank has been created in a development process that was strongly interleaved with the construction of a high-quality syntactic parser, which obeys the same annotation standards. This parser can not only be used to suggest annotations but also to highlight potential problems within them. Such an environment greatly facilitates the extension of the treebank with comparatively modest human effort.

The collection is comprised of three kinds of sentences, which differ in the degree of human revision effort spent on them. While a large part of the corpus has been semi-automatically cross-checked for linguistic plausibility and annotation consistency, others have only been obtained in a fully automatic manner or inspected manually. With these different subsets, the treebank lends itself particularly to experiments on training models with different amounts of data and different levels of quality.

The benefit of using such a large corpus has been demonstrated by evaluating different dependency parsers on it. The parsing quality achieved on the HDT is higher than for any other dependency treebank, and our results show that parsers benefit from increasing the amount of training data even if the original amount was already fairly large. Interestingly, the impact of cross-checking the treebank with the DECCA tools is almost negligible if only the parsing results are considered. Manually correcting the automatically generated annotations, however, gives a huge benefit: A parser trained on automatically parsed sentences commits about twice as many errors as a parser trained on manually corrected data.

Acknowledgements

We would like to thank the publishing house Heise Zeitschriften Verlag for granting permission to use and distribute data from its news archive for scientific purposes.

8. References

Ballesteros, M., Herrera, J., Francisco, V., and Gervás, P. (2012). Are the existing training corpora unnecessarily large? *Procesamiento del Lenguaje Natural*, 48:21--27.

Bohnet, B. (2010). Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89--97, Beijing, China.

Boyd, A., Dickinson, M., and Meurers, D. (2008). On detecting errors in dependency treebanks. *Research on Language and Computation*, 6(2):113--137.

Brants, S., Dipper, S., Eisenberg, P., Hansen-Schirra, S., König, E., Lezius, W., Rohrer, C., Smith, G., and Uszkoreit, H. (2004). TIGER: Linguistic interpretation of a German corpus. *Research on Language and Computation*, 2(4):597--620.

Buchholz, S. and Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149--164, New York City.

Carreras, X. (2007). Experiments with a higher-order projective dependency parser. In *Proceedings of the CoNLL Shared Task Sessioin of EMNLP-CoNLL 2007*, pages 957--961, Prague, Czech Republic.

Foth, K. A., Menzel, W., and Schröder, I. (2000). A transformation-based parsing technique with anytime properties. In *4th Int. Workshop on Parsing Technologies, IWPT-2000*, pages 89 -- 100, Trento, Italy.

Foth, K. A., Daum, M., and Menzel, W. (2004). Interactive grammar development with WCDG. In *The Companion Volume to the Proceedings of 42st Annual Meeting of the Association for Computational Linguistics*, pages 122--125, Barcelona, Spain.

Foth, K. A. (2006a). Eine umfassende Constraint-Dependenz-Grammatik des Deutschen. http://nats-www.informatik.uni-hamburg.de/pub/CDG/CdgManuals/deutsch.pdf.

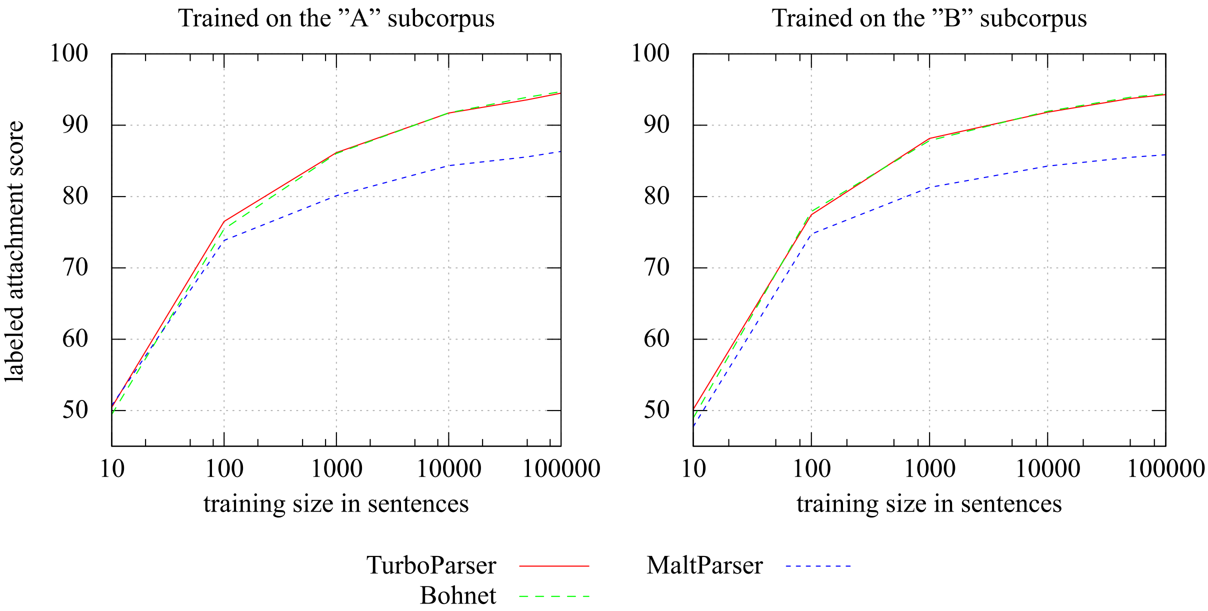

Figure 3: Results for the parts with manual correction

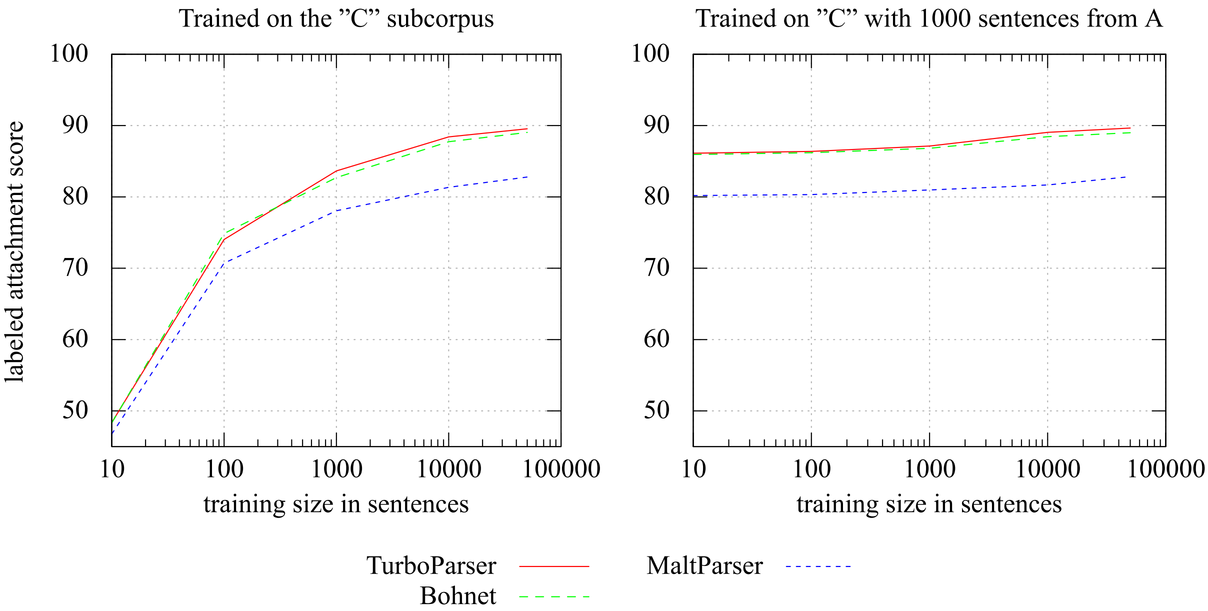

Figure 4: Results for the part without manual correction

Foth, K. A. (2006b). *Hybrid Methods of Natural Language Analysis*. Ph.D. thesis, Universität Hamburg.

Gómez-Rodríguez, C. and Nivre, J. (2010). A transition-based parser for 2-planar dependency structures. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1492--1501, Uppsala, Sweden.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1--18, Boulder, Colorado. Association for Computational Linguistics.

Havelka, J. (2007). Beyond projectivity: Multilingual evaluation of constraints and measures on non-projective structures. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 608--615, Prague, Czech Republic.

Huang, L. and Sagae, K. (2010). Dynamic programming for linear-time incremental parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1077--1086, Uppsala, Sweden.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. (1994). The Penn Treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114--119, Stroudsburg, PA, USA. Association for Computational Linguistics.

Martins, A., Smith, N., and Xing, E. (2009). Concise integer linear programming formulations for dependency parsing. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 342--350, Suntec, Singapore.

Martins, A., Almeida, M., and Smith, N. A. (2013). Turning on the turbo: Fast third-order non-projective turbo parsers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617--622, Sofia, Bulgaria.

McDonald, R. T. and Pereira, F. C. N. (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

McDonald, R., Pereira, F., Ribarov, K., and Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523--530, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., and Yuret, D. (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915--932.

Nivre, J. (2003). An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT 03*.

Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Universität Stuttgart / Universität Tübingen.

Schröder, I. (2002). *Natural Language Parsing with Graded Constraints*. Ph.D. thesis, Universität Hamburg.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159--177, Manchester, England.

Telljohann, H., Hinrichs, E., and Kübler, S. (2004). The Tüba-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229--2235.