DeepBank: A Dynamically Annotated Treebank of the Wall Street Journal

Dan Flickinger[†], Yi Zhang[‡♯], Valia Kordoni[◇♯]
† CSLI, Stanford University
‡ Department of Computational Linguistics, Saarland University, Germany
♯ LT-Lab, German Research Center for Artificial Intelligence
◇ Department of English Studies, Humboldt University of Berlin

E-mail: danf@stanford.edu, yizhang@dfki.de,
kordonie@anglistik.hu-berlin.de

Abstract

This paper describes a large on-going effort, nearing completion, which aims to annotate the text of all of the 25 Wall Street Journal sections included in the Penn Treebank, using a hand-written broad-coverage grammar of English, manual disambiguation, and a PCFG approximation for the sentences not yet successfully analyzed by the grammar. These grammar-based annotations are linguistically rich, including both fine-grained syntactic structures grounded in the Head-driven Phrase Structure Grammar framework, as well as logically sound semantic representations expressed in Minimal Recursion Semantics. The linguistic depth of these annotations on a large and familiar corpus should enable a variety of NLP-related tasks, including more direct comparison of grammars and parsers across frameworks, identification of sentences exhibiting linguistically interesting phenomena, and training of more accurate robust parsers and parse-ranking models that will also perform well on texts in other domains.

1 Introduction

This paper presents the English DeepBank, an on-going project whose aim is to produce rich syntactic and semantic annotations for the 25 Wall Street Journal (WSJ) sections included in the Penn Treebank (PTB: [16]). The annotations are for the most part produced by manual disambiguation of parses licensed by the English Resource Grammar (ERG: [10]), which is a hand-written, broad-coverage grammar for English in the framework of Head-driven Phrase Structure Grammar (HPSG: [19]).

Large-scale full syntactic annotation has for quite some time been approached with mixed feelings by researchers. On the one hand, detailed syntactic annotation can serve as a basis for corpus-linguistic study and improved data-driven NLP

methods. When combined with supervised machine learning methods, such richly annotated language resources including treebanks play a key role in modern computational linguistics. The availability of large-scale treebanks in recent years has contributed to the blossoming of data-driven approaches to robust and practical parsing.

On the other hand, the creation of detailed and consistent syntactic annotations on a large scale turns out to be a challenging task.[1] From the choice of the appropriate linguistic framework and the design of the annotation scheme to the choice of the text source and the working protocols on the synchronization of the parallel development, as well as quality assurance, each of the steps in the entire annotation procedure presents non-trivial challenges that can impede the successful production of such resources.

The aim of the DeepBank project is to overcome some of the limitations and shortcomings which are inherent in manual corpus annotation efforts, such as the German Negra/Tiger Treebank ([2]), the Prague Dependency Treebank ([11]), and the TüBa-D/Z.[2] All of these have stimulated research in various sub-fields of computational linguistics where corpus-based empirical methods are used, but at a high cost of development and with limits on the level of detail in the syntactic and semantic annotations that can be consistently sustained. The central difference in the DeepBank approach is to adopt the *dynamic* treebanking methodology of Redwoods [18], which uses a grammar to produce full candidate analyses, and has human annotators disambiguate to identify and record the correct analyses, with the disambiguation choices recorded at the granularity of constituent words and phrases. This localized disambiguation enables the treebank annotations to be repeatedly refined by making corrections and improvements to the grammar, with the changes then projected throughout the treebank by reparsing the corpus and re-applying the disambiguation choices, with a relatively small number of new disambiguation choices left for manual disambiguation.

For the English DeepBank annotation task, we make extensive use of resources in the DELPH-IN repository[3], including the PET unification-based parser ([4]), the ERG plus a regular-expression preprocessor ([1]), the LKB grammar development platform ([7]), and the [incr tsdb()] competence and performance profiling system ([17]), which includes the treebanking tools used for disambiguation and inspection. Using these resources, the task of treebank construction shifts from a labor-intensive task of drawing trees from scratch to a more intelligence-demanding task of choosing among candidate analyses to either arrive at the desired analysis or reject all candidates as ill-formed. The DeepBank approach should be differentiated from so-called treebank conversion approaches, which derive a new treebank

[1] Besides[18], which we draw more on for the remainder of the paper, similar work has been done in the HPSG framework for Dutch [22]. Moreover, there is quite a lot of related research in the LFG community, e.g., in the context of the ParGram project: [9] for German, [14] for English, and the (related) Trepil project, e.g., [20] for Norwegian.

[2] http://www.sfs.nphil.uni-tuebingen.de/en_tuebadz.shtml

[3] http://www.delph-in.net

from another already existing one, such as the Penn Treebank, mapping from one format to another, and often from one linguistic framework to another, adapting and often enriching the annotations semi-automatically. In contrast, the English DeepBank resource is constructed by taking as input only the original 'raw' WSJ text, sentence-segmented to align with the segmentation in the PTB for ease of comparison, but making no reference to any of the PTB annotations, so that we maintain a fully independent annotation pipeline, important for later evaluation of the quality of our annotations over held-out sections.

2 DeepBank

The process of DeepBank annotation of the Wall Street Journal corpus is organised into iterations of a cycle of parsing, treebanking, error analysis and grammar/treebank updates, with the goal of maximizing the accuracy of annotation through successive refinement.

Parsing Each section of the WSJ corpus is first parsed with the PET parser using the ERG, with lexical entries for unknown words added on the fly based on a conventional part-of-speech tagger, TnT [3]. Analyses are ranked using a maximum-entropy model built using the TADM [15] package, originally trained on out-of-domain treebanked data, and later improved in accuracy for this task by including a portion of the emerging DeepBank itself for training data. A maximum of 500 highest-ranking analyses are recorded for each sentence, with this limit motivated both by practical constraints on data storage costs for each parse forest and by the processing capacity of the [incr tsdb()] treebanking tool. The existing parse-ranking model has proven to be accurate enough to ensure that the desired analysis is almost always in these top 500 readings if it is licensed by the grammar at all. For each analysis in each parse forest, we record the exact derivation tree, which identifies the specific lexical entries and the lexical and syntactic rules applied to license that analysis, comprising a complete 'recipe' sufficient to reconstruct the full feature structure given the relevant version of the grammar. This approach enables relatively efficient storage of each parse forest without any loss of detail.

Treebanking For each sentence of the corpus, the parsing results are then manually disambiguated by the human annotators, using the [incr tsdb()] treebanking tool which presents the annotator with a set of binary decisions, called *discriminants*, on the inclusion or exclusion of candidate lexical or phrasal elements for the desired analysis. This discriminant-based approach of [6] enables rapid reduction of the parse forest to either the single desired analysis, or to rejection of the whole forest for sentences where the grammar has failed to propose a viable analysis.[4] On average, given n candidate trees, $\log_2 n$ decisions are needed in order to

[4]For some sentences, an annotator may be unsure about the correctness of the best available analysis, in which case the analysis can still be recorded in the treebank, but with a lower 'confidence'

fully disambiguate the parse forest for a sentence. Given that we set a limit of 500 candidate readings per sentence, full disambiguation of a newly parsed sentence averages no more than 9 decisions, which enables a careful annotator to sustain a treebanking rate of 30 to 50 sentences per hour on the first pass through the corpus.

Error analysis During the course of this annotation effort, several annotators have been trained and assigned to carry out the initial treebanking of portions of the WSJ corpus, with most sections singly annotated. On successive passes through the treebank, two types of errors are identified and dealt with: mistakes or inconsistencies of annotation, and shortcomings of the grammar such that the desired analysis for a given sentence was not yet available in the parse forest. Errors in annotation include mistakes in constituent boundaries, in lexical choice such as verb valency or even basic part of speech, and in phrasal structures such as the level of attachment of modifiers or the grouping of conjuncts in a coordinated phrase. Our calculation of the inter-annotator agreeemnt using the Cohen's KAPPA[5] on the constituents of the derivation trees after the initial round of treebanking shows a moderate agreement level at $\kappa = 0.6$. Such disagreeements are identified for correction both by systematic review of the recorded 'correct' trees section by section, and by searching through the treebank for specific identifiers of constructions or lexical entries known to be relatively rare in the WSJ, such as the rules admitting questions or imperative clauses.

Shortcomings of the grammar are identified by examining sentences for which annotators did not record a correct analysis, either because no analysis was assigned, or because all of the top 500 candidate analyses were flawed. Some of the sources of error emerge quickly from even cursory analysis, such as the initial absence of a correct treatment in the ERG for measure phrases used as verbal modifiers, which are frequent in the WSJ corpus, as in *the index rose 20 points* or *the market fell 14%*. Other types of errors required more detailed analysis, such as missing lexical entries for some nouns taking verbal complements, as in *the news that Smith was hired* or *the temptation to spend the money*. These fine-grained lexical entries are not correctly predicted on the fly using the part-of-speech tagger, and hence must be added to the 35,000-entry manually supplied lexicon in the ERG.

Grammar & Treebank Update While grammar development proceeds independent of the initial treebank annotation process, we have periodically incorporated improvements to the grammar into the treebank annotation cycle. When a grammar update is incorporated, the treebank also gets updated accordingly by (i) parsing anew all of the sentences in the corpus using the new grammar; (ii) reapplying the recorded annotation decisions; and (iii) annotating those sentences which are not fully disambiguated after step ii, either because new ambiguity was introduced by the grammar changes, or because a sentence which previously failed

score assigned, so the annotation can be reviewed in a later cycle of updates.

to parse now does. The extra manual annotation effort in treebank update is relatively small when compared to the first round of annotation, typically requiring one or two additional decisions for some 5–10% of the previously recorded correct analyses, and new annotation for previously rejected items, which were another 15% of the total in the second round, and much less in successive rounds. Hence these later rounds of updating the treebank proceed more quickly than the initial round of annotation.

Correcting errors of both classes based on analysis of the first pass through DeepBank annotation has resulted in a significant improvement in coverage and accuracy for the ERG over the WSJ corpus. Raw coverage has risen by some 10% from the first pass and the 'survival' rate of successfully treebanked sentences has risen even more dramatically to more than 80% of all sentences in the first 16 sections of the WSJ that have now gone through two rounds of grammar/treebank updates. The table below shows the current status of these first 16 sections of the English DeepBank in terms of "Observed" and "Verified" coverage, where the former reports the number of sentences that received at least one analysis from the ERG, and the latter gives the number of sentences for which the annotator recorded a correct analysis.

Table 1: English DeepBank ERG results for WSJ Sections 00–15

Section	Number of items	Observed coverage	Verified coverage
00	1922	92.2%	82.0%
01	1997	92.3%	81.6%
02	1996	92.3%	84.0%
03	1482	92.0%	82.1%
04	2269	92.6%	81.5%
05	2137	92.3%	81.8%
06	1835	91.3%	81.1%
07	2166	91.9%	82.6%
08	478	90.6%	80.1%
09	2073	92.0%	81.2%
10	1945	91.8%	81.3%
11	2237	91.5%	80.4%
12	2124	94.2%	85.1%
13	2481	94.8%	85.8%
14	2182	94.0%	86.0%
15	2118	94.1%	86.4%
Subtotal	31442	92.6%	82.6%

These figures, which are surprisingly stable across sections both in raw parsing coverage and in treebanked items, show that roughly 18% of the sentences in the corpus fail to receive a correct analysis from the ERG; we discuss the DeepBank annotations for this portion of the corpus in section 4. Note that most of the remaining portion of the WSJ corpus has now been treebanked the first time through, and we expect the remaining updated sections to be completed by the end of the year, excluding three held-out sections reserved for future testing.

3 Annotation formats

In comparison to existing large-scale treebanks, DeepBank stands out as a unique resource which incorporates both syntactic and semantic annotations in a uniform grammar framework. To facilitate the easy access of various layers of annotation in the treebank, multiple formats will be provided in the release of the English DeepBank. The [incr tsdb()] profiles are comprehensive relational databases that record the original ERG derivation trees together with the semantic representations natively expressed in Minimal Recursion Semantics (MRS: [8]) structures. The database also keeps the history of the manual annotations (the disambiguation discriminants). For users interested in simpler or more conventional representations, the HPSG derivations are also converted to PTB-style phrase structure tree representations which employ a mapping of HPSG categories to a smaller set of POS and phrasal categories that roughly corresponding to those of the English PTB. Furthermore, the treebank is also available in a dependency-oriented representation following the format of the CoNLL-2008 Shared Task [21]. The syntactic dependencies are extracted from the derivation trees of the ERG, while the semantic dependencies offer a simplified view of the native MRS structures[12]. It should be noted that not all linguistic information in the native DeepBank annotations is preserved in the PTB phrase structure and CoNLL dependency formats. Nevertheless, they offer easy access to the data in familiar formats.

We give an example of each of these annotations for a simple sentence from the corpus, beginning with the native derivation which contains sufficient information to enable full reconstruction of the HPSG feature structure returned by the parser. Next is the simplified PTB-style labeled bracketing for the example, then the native semantic representation in MRS, and finally the CoNLL-style dependency view of the syntax and the semantics.

```
(root_strict
 (sb-hd_mc_c
  (hdn_bnp_c
   (aj-hdn_norm_c
    (j-j_crd-att-t_c
     (v_j-nb-pas-tr_dlr (v_pas_odlr (estimate_v4 ("estimated"))))
     (mrk-nh_evnt_c
      (and_conj ("and"))
      (actual_a1 ("actual"))))
    (hdn-aj_rc_c
     (hdn_optcmp_c (n_pl_olr (result_n1 ("results"))))
     (vp_rc-redrel_c
      (hd-cmp_u_c
       (v_prp_olr (involve_v2 ("involving")))
       (hdn_bnp_c (hdn_optcmp_c (n_pl_olr (loss_n1 ("losses"))))))))))
  (hd-cmp_u_c
   (be_c_are ("are"))
   (hd_optcmp_c (w_period_plr (v_pas_odlr (omit_v1 ("omitted."))))))))
```

Figure 1: Sample DeepBank native derivation tree for "*Estimated and actual results involving losses are omitted.*"

In the derivation show in Figure 1, we see that a combination of very general rules and construction-specific ones have been applied to license this analysis: the rule that combines any head with a complement (the *hd-cmp_u_c* rule) is used for the verb phrase "involving losses" and again for "are omitted", while the narrowly constrained rule that converts a VP into a post-nominal modifier (the *vp_rc-redrel_c* rule) is used to ensure the correct semantics for the nominal phrase "results involving losses". The specific lexical entry identifiers are also included in the derivation, showing for example that the entry used for "estimated" here is *estimate_v4*, which happens to be the simple transitive verb, not, say, the raising verb that would be needed for *we estimated there to be dozens of applicants*.

```
(S
 (NP (N (AP (AP (V estimated))
             (AP (CONJ and)
                 (AP actual)))
        (N (N results)
           (S (VP (V involving)
                  (NP (N losses)))))))
 (VP (V are)
     (VP (V omitted.)))))
```

Figure 2: Sample DeepBank PTB-style labeled bracketing for "*Estimated and actual results involving losses are omitted.*"

The simplified view of the syntactic analysis in Figure 2 employs one of a small set of familiar lexical and phrasal category labels for each bracketed constituent. These node labels can be helpful both for cross-framework parser comparisons, and also for coarse-grained searches of the treebank, such as when looking for all noun phrases in a certain configuration, ignoring the internal composition of each NP.

```
<h1,e3:prop:pres:indicative:-:-,
{h4:udef_q<0:45>(x6, h5, h7),
h8:_estimate_v_at<0:9>(e9, i10, x6),
h8:parg_d<0:9>(e11, e9, x6),
h12:_and_c<10:13>(e13, h8, e9, h14, e15),
h14:_actual_a_1<14:20>(e15, x6),
h12:_result_n_of<21:28>(x6:3:pl:+, i16),
h12:_involve_v_1<29:38>(e17, x6, x18),
h19:udef_q<39:45>(x18, h20, h21),
h22:_loss_n_of<39:45>(x18:3:pl:+, i23),
h2:_omit_v_1<50:58>(e3, i24, x6),
h2:parg_d<50:58>(e25, e3, x6)},
{h1 qeq h2, h5 qeq h12, h20 qeq h22}>
```

Figure 3: Sample DeepBank semantics in native MRS representation for "*Estimated and actual results involving losses are omitted.*"

The compact view of the MRS representation shown in Figure 3 employs a strict ascending ordering convention on the arguments for each elementary predication, with the first argument being the inherent variable (a referential index for nominal predications such as *_loss_n_of* and an event variable otherwise). Thus the verbal

ID	FORM	LEMMA	GPOS	HEAD	DEPREL	PRED	ARGS-P1	ARGS-P2	ARGS-P3	ARGS-P4	ARGS-P5	ARGS-P6	ARGS-P7
1	Estimated	estimate	v_np	4	aj-hdn_norm	_estimate_v_at	ARG0	L-INDEX	_	_	_	_	_
2	and	and	c_xp_and	1	j-j_crd-att-t	_and_c	_	ARG0	_	_	_	_	_
3	actual	actual	aj_-_i	2	mrk-nh_evnt	_actual_a_1	_	R-INDEX	ARG0	_	_	_	_
4	results	result	n_pp_c-ns-of	7	sb-hd_mc	_result_n_of	ARG2	_	ARG1	ARG0	ARG1	_	ARG2
5	involving	involve	v_np	4	hdn-aj_rc	_involve_v_1	_	_	_	_	ARG0	_	_
6	losses	loss	n_pp_mc-of	5	hd-cmp_u	_loss_n_of	_	_	_	_	ARG2	ARG0	_
7	are	be	v_prd_are	0	root_strict	_	_	_	_	_	_	_	_
8	omitted.	omit	v_np	7	hd-cmp_u	_omit_v_1	_	_	_	_	_	_	ARG0

Figure 4: Sample DeepBank CoNLL-style dependencies for "*Estimated and actual results involving losses are omitted.*"

predication *_omit_v_1* introduced by the passive "omitted" only has its ARG2 instantiated with the index introduced by "results", leaving its ARG1 uninstantiated, as indicated by the presence of an "i" rather than an "x" variable as the second of its three arguments. Each predication is also marked with a character span from the original sentence, linking this component of the semantics to the corresponding word or phrase that introduced it.

ID	FORM	LEMMA	GPOS	HEAD	DEPREL	PRED	ARGS-P1	ARGS-P2	ARGS-P3	ARGS-P4
1	Estimated	estimate	VBN	4	NMOD	estimate.01	_	AM-ADV	_	_
2	and	and	CC	1	COORD	_	_	_	_	_
3	actual	actual	JJ	2	CONJ	_	_	_	_	_
4	results	result	NNS	7	SBJ	result.01	A1	A2	A2	A1
5	involving	involve	VBG	4	APPO	involve.01	_	_	_	_
6	losses	losses	NNS	5	OBJ	_	_	_	A1	_
7	are	be	VBP	0	ROOT	_	_	_	_	_
8	omitted	omit	VBN	7	VC	omit.01	_	_	_	_
9	.	.	.	7	P	_	_	_	_	_

Figure 5: Sample of original CoNLL (2008) dependencies derived from PTB and PropBank/NomBank annotation

The CoNLL-style dependency format shown in Figure 4 incorporates the essential syntactic and semantic structures of the HPSG analysis in a uniformed token-based dependency representation[5]. The GPOS field contains the "golden" lexical type selected for the corresponding token. The HEAD field records the token ID of the dependency head. The DEPREL is the corresponding dependency type which is inherited from the HPSG rule name. The PRED field contains the name of the elementary predications from the MRS (hence not limited to verbal and nominal predicates). The remaining ARGS fields identify the arguments of each predicate.

In comparison to the PTB + PropBank/NomBank derived dependency annotation for CoNLL Shared Task 2008 (see Figure 5 for an example), the DeepBank data in CoNLL format offers more fine-grained POS and dependency types, and more densely populated semantic graphs. For example, in comparison to the dependency type inventory of [13] used in the CoNLL Shared Tasks which does not distinguish different types of nominal modifiers (NMOD), our dependencies further mark such head-adjunct relations by the type of the modifier being a pre-head adjunct (*aj-hdn_adjn*, as in *"The [big old cat] slept."*), a post-head relative clause (*hdn-aj_rc*, as in *"The [cat we chased] ran."*), or a post-head reduced relative clause (*hdn-aj_redrel*, as in *"A [cat in a tree] fell."*)

[5]Due to the limited page width, not all the columns in the CoNLL 2008 format are shown here.

4 Patching Coverage Gaps with An Approximating PCFG

As we noted above, one potential criticism against a purely grammar-based treebanking approach addresses its lack of complete coverage in analyzing all sentences in the corpus. The missing gaps in coverage are due to one or more of three causes: (i) ill-formed texts as input to the grammar (rare but present); (ii) the lack of linguistic coverage in the grammar implementation (most frequent); or (iii) limits on computing resources – time or memory – imposed in the analysis of any one sentence (perhaps 20% of the failed parses). The first issue is not specific to grammar-based treebanking, and in fact, manual treebanking projects also carefully select (and in many cases edit) the texts to be annotated. The top criterion for the selection step is to keep the meaningful and representative texts while discarding the problematic items for which full linguistic annotation is not worthwhile. For the second and third issues of either incomplete grammar coverage or the lack of efficiency in processing, there is legitimate concern over the robustness of deep linguistic grammars such as the ERG in comparison to creative and flexible human annotators.

In our discriminant-based approach of treebanking, the coverage gap shows up in two ways: either the grammar fails to parse a specific input utterance, or all the candidate analyses proposed by the grammar are rejected through the manual disambiguation step. Both suggest that a desired analysis is missing due to certain constraints in the grammar. Our experience with the Wall Street Journal corpus and the ERG shows that about 8% of the sentences fail to parse, while another 10% received no acceptable analysis despite getting one or more parses from the ERG. In both cases, using the discriminant-based treebanking tools, annotators cannot record an existing good tree for the sentence.

To annotate the sentences in the grammar coverage gap, we use a robust and overgenerating grammar that approximates the parsing behavior of the ERG. More specifically, an approximating probabilistic context-free grammar (PCFG) is extracted from the automatically parsed treebank of the ERG. The categories in the PCFG are the HPSG rule names annotated with additional information either from the syntactic context (derivation tree) or the detailed properties in the feature structure. Due to the unrestrictive nature of the PCFG, it achieves almost full coverage on all the sentences from the original corpus. The approximating PCFG delivers the most likely pseudo-derivations of ERG according to a generative probabilistic model. In combination with the feature structures of the rules and the lexical entries from the orignal ERG, we can recompose the semantics by doing unification on these derivations. In cases where the unification fails, a robust unifier is called instead to override one side of the conflicting constraints according to certain heuristics.

The evaluation shown in [23] suggests that this PCFG, with careful selection of addiontal annotations and the massive automatically created training treebank, achieves very good parsing accuracy. When tested on sentences that the ERG covers correctly, the best PCFG achieved 84.9 (syntactic) ParsEval F1 score, and 84.2 F1 in the semantic argument relation evaluation (EDM_A). Both measures are about

2% lower than the HPSG parser with ERG. The PCFG succeeds in parsing over 99% of the test set, while the original ERG successfully covers about 80% of it. In a comparison to the parsing accuracy of the state-of-the-art Berkeley parser trained with the same corpus, our PCFG training was much more scalable (with up to 50 million automatically ERG parsed trees), yielding much better overall accuracy.

Lastly, we have developed a graphical tree editor that allows the annotators to manually correct the remaining errors in the PCFG parses. The tool not only supports an intuitive drag-and-drop style of editing, but also records the entire editing sequence, creating additional raw annotation data for future research. Preliminary experience on the post-editing steps suggests that an annotator can correct 35-40 sentences per hour, producing for each a derivation tree which contains at least one constituent not (yet) licensed by the ERG, but necessary for the correct analysis of the sentence.

5 Next Steps

Among the principal advantages claimed for this DeepBank approach is the ability to make successive refinements to the treebank annotations, by making changes to the grammar or to the parsing configuration, and then reparsing and updating with the existing discriminant-based annotations. One planned change in that parsing configuration is to record in the database the full (packed) parse forest for each sentence, rather than the 500 highest-ranked parses currently stored. Manual disambiguation from the full forest will require a new treebanking tool, still under development, but initial experiments already confirm that the existing discriminants are sufficient to automatically fully disambiguate the great majority of the previously treebanked WSJ sentences even working with full parse forests. This full-forest method will provide greater stability in the English DeepBank, eliminating the current minor but annoying uncertainty that results from the dependence on parse ranking to preserve the desired analysis among the top-ranked 500.

6 Conclusion

The English DeepBank provides linguistically rich syntactic and semantic annotations grounded in a well-established and leading linguistic theory (HPSG) for a large and familiar corpus, the million-word Wall Street Journal portion also annotated in the Penn Treebank. The first public release of this resource will include manually selected full analyses produced by the English Resource Grammar for more than 80% of these 50,000 sentences, providing unmatched consistency and linguistic detail, available in multiple formats and representations. The remainder of the corpus will be annotated with compatible though approximate syntactic and semantic analyses produced using a PCFG trained on the manually annotated treebank, to ensure complete coverage of the corpus in the treebank. Adopting the Redwoods methodology for constructing and maintaining this dynamic treebank

will enable further improvements in the grammar to be projected into updated versions of the DeepBank, along with correction of any remaining annotation errors. We believe that an annotated resource of this scale for this corpus will be useful for research both in NLP and in corpus-based theoretical work in linguistics and psycholinguistics.

Acknowledgements

The project is partially supported by the Erasmus Mundus European Masters Program in Language and Communication Technologies (http://www.lct-master.org; EM Grant Number: 2007-0060). The second author also thanks the *Deependance* project funded by BMBF (01IW11003) for its support of the work.

References

[1] Peter Adolphs, Stephan Oepen, Ulrich Callmeier, Berthold Crysmann, Dan Flickinger, and Bernd Kiefer. Some fine points of hybrid natural language parsing. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008.

[2] Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. The tiger treebank. In *Proceedings of the workshop on treebanks and linguistic theories*, pages 24–41, 2002.

[3] Thorsten Brants. TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP 2000)*, Seattle, USA, 2000.

[4] Ulrich Callmeier. Efficient parsing with large-scale unification grammars. Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 2001.

[5] Jean Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[6] David Carter. The treebanker: a tool for supervised training of parsed corpora. In *Proceedings of the ACL Workshop on Computational Environments for Grammar Development and Linguistic Engineering*, pages 9–15, Madrid, Spain, 1997.

[7] Ann Copestake. *Implementing Typed Feature Structure Grammars*. CSLI, Stanford, USA, 2002.

[8] Ann Copestake, Dan Flickinger, Carl J. Pollard, and Ivan A. Sag. Minimal recursion semantics: an introduction. *Research on Language and Computation*, 3(4):281–332, 2005.

[9] Stefanie Dipper. Grammar-based corpus annotation. In Anne AbeillÃ©, Thorsten Brants, and Hans Uszkoreit, editors, *Proceedings of the Workshop on Linguistically Interpreted Corpora LINC-2000 , Luxembourg*, pages 56–64, 2000.

[10] Dan Flickinger. On building a more efficient grammar by exploiting types. In Stephan Oepen, Dan Flickinger, Jun'ichi Tsujii, and Hans Uszkoreit, editors, *Collaborative Language Engineering*, pages 1–17. CSLI Publications, 2002.

[11] Jan Hajič, Alena Böhmová, Eva Hajičová, and Barbora Vidová-Hladká. The Prague Dependency Treebank: A Three-Level Annotation Scenario. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer, 2000.

[12] Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. Who did what to whom? a contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 2–11, Jeju, Republic of Korea, 2012.

[13] Richard Johansson and Pierre Nugues. Extended Constituent-to-dependency Conversion for English. In *Proceedings of NODALIDA 2007*, Tartu, Estonia, 2007.

[14] Tracy Holloway King, Richard Crouch, Stefan Riezler, Mary Dalrymple, and Ronald M. Kaplan. The parc 700 dependency bank. In *In Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC-03*, pages 1–8, 2003.

[15] Robert Malouf, John Carroll, and Ann Copestake. Efficient feature structure operations without compilation. *Natural Language Engineering*, 6:29–46, 2000.

[16] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[17] Stephan Oepen. [incr tsdb()] — competence and performance laboratory. User manual. Technical report, Computational Linguistics, Saarland University, Saarbrücken, Germany, 2001.

[18] Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. The LinGO Redwoods treebank: motivation and preliminary applications. In *Proceedings of COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes*, Taipei, Taiwan, 2002.

[19] Carl J. Pollard and Ivan A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago, USA, 1994.

[20] Victoria Rosén, Paul Meurer, and Koenraad de Smedt. LFG Parsebanker: A Toolkit for Building and Searching a Treebank as a Parsed Corpus. In Frank Van Eynde, Anette Frank, Gertjan van Noord, and Koenraad De Smedt, editors, *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories (TLT7)*, pages 127–133, Utrecht, 2009. LOT.

[21] Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*, Manchester, UK, 2008.

[22] L. van der Beek, G. Bouma, R. Malouf, and G. van Noord. The alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN) 2001*, Twente University, 2002.

[23] Yi Zhang and Hans-Ulrich Krieger. Large-scale corpus-driven pcfg approximation of an hpsg. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 198–208, Dublin, Ireland, 2011.