The LinGO Redwoods Treebank
Motivation and Preliminary Applications

Stephan Oepen, Kristina Toutanova, Stuart Shieber,
Christopher Manning, Dan Flickinger, and Thorsten Brants
{oe|kristina|manning|dan}@csli.stanford.edu,
shieber@deas.harvard.edu, brants@parc.xerox.com

Abstract

The LinGO Redwoods initiative is a seed activity in the design and development of a new type of treebank. While several medium- to large-scale treebanks exist for English (and for other major languages), pre-existing publicly available resources exhibit the following limitations: (i) annotation is mono-stratal, either encoding topological (phrase structure) *or* tectogrammatical (dependency) information, (ii) the depth of linguistic information recorded is comparatively shallow, (iii) the design and format of linguistic representation in the treebank hard-wires a small, predefined range of ways in which information can be extracted from the treebank, and (iv) representations in existing treebanks are static and over the (often year- or decade-long) evolution of a large-scale treebank tend to fall behind the development of the field. LinGO Redwoods aims at the development of a novel treebanking methodology, *rich* in nature and *dynamic* both in the ways linguistic data can be retrieved from the treebank in varying granularity and in the constant evolution and regular updating of the treebank itself. Since October 2001, the project is working to build the foundations for this new type of treebank, to develop a basic set of tools for treebank construction and maintenance, and to construct an initial set of 10,000 annotated trees to be distributed together with the tools under an open-source license.

1 Why Another (Type of) Treebank?

For the past decade or more, symbolic, linguistically oriented methods and statistical or machine learning approaches to NLP have often been perceived as incompatible or even competing paradigms. While shallow and probabilistic processing techniques have produced useful results in many classes of applications, they have not met the full range of needs for NLP, particularly where precise interpretation is important, or where the variety of linguistic expression is large relative to the amount of training data available. On the other hand, deep approaches to NLP have only recently achieved broad enough grammatical coverage *and* sufficient processing efficiency to allow the use of precise linguistic grammars in certain types of real-world applications.

In particular, applications of broad-coverage analytical grammars for parsing or generation require the use of sophisticated statistical techniques for resolving ambiguities; the transfer of Head-Driven Phrase Structure Grammar (HPSG) systems into industry, for example, has amplified the need for general parse ranking, disambiguation, and robust recovery techniques. We observe general consensus on the necessity for bridging activities, combining symbolic and stochastic approaches to NLP. But although we find promising research in stochastic parsing in a number of frameworks, there is a lack of appropriately rich and dynamic language corpora for HPSG. Likewise, stochastic parsing has so far been focussed on information-extraction-type applications and lacks any depth of semantic interpretation. The Redwoods initiative is designed to fill in this gap.

In the next section, we present some of the motivation for the LinGO Redwoods project as a treebank development process. Although construction of the treebank is in its early stages, we present in Section 3 some preliminary results of using the treebank data already acquired on concrete applications. We show, for instance, that even simple statistical models of parse ranking trained on the Redwoods corpus built so far can disambiguate parses with close to 80% accuracy.

2 A Rich and Dynamic Treebank

The Redwoods treebank is based on open-source HPSG resources developed by a broad consortium of research groups including researchers at Stanford (USA), Saarbrücken (Germany), Cambridge, Edinburgh, and Sussex (UK), and Tokyo (Japan). Their wide distribution and common acceptance make the HPSG framework and resources an excellent anchor point for the Redwoods treebanking initiative.

The key innovative aspect of the Redwoods approach to treebanking is the anchoring of all linguistic data captured in the treebank to the HPSG framework and a generally-available broad-coverage grammar of English, the LinGO English Resource Grammar (Flickinger, 2000) as implemented with the LKB grammar development environment (Copestake, 2002). Unlike existing treebanks, there is no need to define a (new) form of grammatical representation specific to the treebank. Instead, the treebank records complete syntacto-semantic analyses as defined by the LinGO ERG and provide tools to extract different types of linguistic information at varying granularity.

The treebanking environment, building on the [incr tsdb()] profiling environment (Oepen & Callmeier, 2000), presents annotators, one sentence at a time, with the full set of analyses produced by the grammar. Using a pre-existing tree comparison tool in the LKB (similar in kind to the SRI Cambridge TreeBanker; Carter, 1997), annotators can quickly navigate through the parse forest and identify the correct or preferred analysis in the current context (or, in rare cases, reject all analyses proposed by the grammar). The tree selection tool presents users, who need little expert knowledge of the underlying grammar, with a range of basic properties that distinguish competing analyses and that are relatively easy to judge. All disambiguating decisions made by annotators are recorded in the [incr tsdb()] database and thus become available for (i) later dynamic extraction from the annotated profile or (ii) dynamic propagation into a more recent profile obtained from re-running a newer version of the grammar on the same corpus.

Important innovative research aspects in this approach to treebanking are (i) enabling users of the treebank to extract information of the type they need and to transform the available representation into a form suited to their needs and (ii) the ability to update the treebank with an enhanced version of the grammar in an automated fashion, viz. by re-applying the disambiguating decisions on the corpus with an updated version of the grammar.

Depth of Representation and Transformation of Information Internally, the [incr tsdb()] database records analyses in three different formats, viz. (i) as a derivation tree composed of identifiers of lexical items and constructions used to build the analysis, (ii) as a traditional phrase structure tree labeled with an inventory of some fifty atomic labels (of the type 'S', 'NP', 'VP' et al.), and (iii) as an underspecified MRS (Copestake, Lascarides, & Flickinger, 2001) meaning representation. While representation (ii) will in many cases be similar to the representation found in the Penn Treebank, representation (iii) subsumes the functor – argument (or tectogrammatical) structure advocated in the Prague Dependency Treebank or the German TiGer corpus. Most importantly, however, representation (i) provides all the information required to replay the full HPSG analysis (using the original grammar and one of the open-source HPSG processing environments, e.g., the LKB or PET, which already have been interfaced to [incr tsdb()]). Using the latter approach, users of the treebank are enabled to extract information in whatever representation they require, simply by reconstructing full analyses and adapting the existing mappings (e.g., the inventory of node labels used for phrase structure trees) to their needs. Likewise, the existing [incr tsdb()] facilities for comparing across competence and performance profiles can be deployed to evaluate results of a (stochastic) parse disambiguation system, essentially using the preferences recorded in the treebank as a 'gold standard' target for comparison.

Automating Treebank Construction Although a precise HPSG grammar like the LinGO ERG will typically assign a small number of analyses to a given sentence, choosing among a few or sometimes a few dozen readings is time-consuming and error-prone. The project is exploring two approaches to automating the disambiguation task, (i) seeding lexical selection from a part-of-speech (POS) tagger and (ii) automated inter-annotator comparison and assisted resolution of conflicts.

Treebank Maintenance and Evolution One of the challenging research aspects of the Redwoods initiative is about developing a methodology for automated updates of the treebank to reflect the continuous evolution of the underlying linguistic framework and of the LinGO grammar. Again building on the notion of elementary linguistic discriminators, we expect to explore the semi-automatic propagation of recorded disambiguating decisions into newer versions of the parsed corpus. While it can be assumed that the basic phrase structure inventory and granularity of lexical distinctions have stabilized to a certain degree, it is not guaranteed that one set of discriminators will always fully disambiguate a more recent set of analyses for the same utterance (as the grammar may introduce new ambiguity), nor that re-playing a history of disambiguating decisions will necessarily identify the correct, preferred analysis for all sentences. A better understanding of the nature of discriminators and relations holding among them is expected to provide the foundations for an update procedure that, ultimately, should be mostly automated, with minimal manual inspection, and which can become part of the regular regression test cycle for the grammar.

Scope and Current State of Seeding Initiative The first 10,000 trees to be hand-annotated as part of the kick-off initiative are taken from a domain for which the English Resource Grammar is known to exhibit broad and accurate coverage, viz. transcribed face-to-face dialogues in an appointment scheduling and travel arrangement domain.[1] For the follow-up phase of the project, it is expected to move into a second domain and text genre, presumably more formal, edited text taken from newspaper text or another widely available on-line source. As of June 2002, the seeding initiative is well underway. The integrated treebanking environment, combining [incr tsdb()] and the LKB tree selection tool, has been established and has been deployed in a first iteration of annotating the VerbMobil utterances. The approach to parse selection through minimal discriminators turned out to be not hard to learn for a second-year Stanford undergraduate in linguistics, and allowed completion of the first iteration in less than ten weeks. Table 1 summarizes the current Redwoods status.

[1] Corpora of some 50,000 such utterances are readily available from the VerbMobil project (Wahlster, 2000) and have already been studied extensively among researchers world-wide.

[2] Of the four data sets only VM32 has been double-checked by an expert grammarian and (almost) completely disambiguated to date; therefore it exhibits an interestingly higher degree of phrasal ambiguity in the 'active = 1' subset.

corpus	total				active = 0				active = 1				active > 1				unannotated			
	♯	‖	⊙	×	♯	‖	⊙	×	♯	‖	⊙	×	♯	‖	⊙	×	♯	‖	⊙	×
VM6	2422	7·7	4·2	32·9	218	8·0	4·4	9·7	1910	7·0	4·0	7·5	80	10·0	4·8	23·8	214	14·9	4·3	287·5
VM13	1984	8·5	4·0	37·9	175	8·5	4·1	9·9	1491	7·2	3·9	7·5	85	9·9	4·5	22·1	233	14·1	4·2	22·1
VM31	1726	6·2	4·5	22·4	164	7·9	4·6	8·0	1360	6·6	4·5	5·9	61	10·1	4·2	14·5	141	13·5	4·7	201·5
VM32	608	7·4	4·3	25·6	51	10·7	4·3	54·4	551	7·9	4·4	19·0	5	12·2	3·9	27·2	1	21·0	6·1	2220·0

Table 1: Redwoods development status as of June 2002: four sets of transcribed and hand-segmented VerbMobil dialogues have been annotated. The columns are, from left to right, the total number of sentences (excluding fragments) for which the LinGO grammar has at least one analysis ('♯'), average length ('‖'), lexical and structural ambiguity ('⊙' and '×', respectively), followed by the last four metrics broken down for the following subsets: sentences (i) for which the annotator rejected all analyses (no active trees), (ii) where annotation resulted in exactly one preferred analysis (one active tree), (iii) those where full disambiguation was not accomplished through the first round of annotation (more than one active tree), and (iv) massively ambiguous sentences that have yet to be annotated.[2]

3 Early Experimental Results

Development of the treebank has just started. Nonetheless, we have performed some preliminary experiments on concrete applications to motivate the utility of the resource being developed. In this section, we describe experiments using the Redwoods treebank to build and test systems for parse disambiguation. As a component, we build a tagger for the HPSG lexical tags in the treebank, and report results on this application as well.

Any linguistic system that allows multiple parses of strings must address the problem of selecting from among the admitted parses the preferred one. A variety of approaches for building statistical models of parse selection are possible. At the simplest end, we might look only at the lexical type sequence assigned to the words by each parse and rank the parse based on the likelihood of that sequence. These lexical types – the preterminals in the derivation – are essentially part-of-speech tags, but encode considerably finer-grained information about the words. Well-understood statistical part-of-speech tagging technology is sufficient for this approach.

In order to use more information about the parse, we might examine the entire derivation of the string. Most probabilistic parsing research – including, for example, work by by Collins (1997), and Charniak (1997) – is based on branching process models (Harris, 1963). The HPSG derivations that the treebank makes available can be viewed as just such a branching process, and a stochastic model of the trees can be built as a probabilistic context-free grammar (PCFG) model. Abney (1997) notes important problems with the soundness of the approach when a unification-based grammar is actually determining the derivations, motivating the use of log-linear models (Agresti, 1990) for parse ranking that Johnson and colleagues further developed (Johnson, Geman, Canon, Chi, & Riezler, 1999). These models can deal with the many interacting dependencies and the structural complexity found in constraint-based or unification-based theories of syntax.

Nevertheless, the naive PCFG approach has the advantage of simplicity, so we pursue it and the tagging approach to parse ranking in these proof-of-concept experiments (more recently, we have begun work on building log-linear models over HPSG signs (Toutanova & Manning, 2002)). The learned models were used to rank possible parses of unseen test sentences according to the probabilities they assign to them. We report parse selection performance as percentage of test sentences for which the correct parse was highest ranked by the model. (We restrict attention in the test corpus to sentences that are ambiguous according to the grammar, that is, for which the parse selection task is nontrivial.) We examine four models: an HMM tagging model, a simple PCFG, a PCFG with grandparent annotation, and a hybrid model that combines predictions from the PCFG and the tagger. These models will be described in more detail presently.

The tagger that we have implemented is a standard trigram HMM tagger, defining a joint probability distribution over the preterminal sequences and yields of these trees. Trigram probabilities are smoothed by linear interpolation with lower-order models. For comparison, we present the performance of a unigram tagger and an upper-bound oracle tagger that knows the true tag sequence and scores highest the parses that have the correct preterminal sequence.

The PCFG models define probability distributions over the trees of derivational types corresponding to the HPSG analyses of sentences. A PCFG model has parameters $\theta_{i,j}$ for each rule $A_i \rightarrow \alpha_j$ in the corresponding context free grammar.[3] In our application, the nonterminals in the PCFG A_i are rules of the HPSG grammar used to build the parses (such as HEAD-COMPL or HEAD-ADJ). We set the parameters to maximize the likelihood of the set of derivation trees for the preferred parses of the sentences in a training set. As noted above, estimating probabilities from local tree counts in the treebank does not provide a maximum likelihood estimate of the observed data, as the grammar rules further constrain the possible derivations. Essentially, we are making an assumption of context-freeness of rule application that does not hold in the case of the HPSG grammar. Nonetheless, we can still build the model and use it to rank parses.

[3] For an introduction to PCFG grammars see, for example, Manning & Schütze (1999).

As previously noted by other researchers (Charniak & Caroll, 1994), extending a PCFG with grandparent annotation improves the accuracy of the model. We implemented an extended PCFG that conditions each node's expansion on its parent in the phrase structure tree. The extended PCFG (henceforth PCFG-GP) has parameters $P(^{A_k}A_i \rightarrow \alpha_j | A_k, A_i)$. The resulting grammar can be viewed as a PCFG whose nonterminals are pairs of the nonterminals of the original PCFG.

The combined model scores possible parses using probabilities from the PCFG-GP model together with the probability of the preterminal sequence of the parse tree according to a trigram tag sequence model. More specifically, for a tree T,

$$Score(t) = \log(P_{PCFG\text{-}GP}(T)) + \lambda \log(P_{TRIG}(tags(T))$$

where $P_{TRIG}(tags(T))$ is the probability of the sequence of preterminals $t_1 \cdots t_n$ in T according to a trigram tag model:

$$P_{TRIG}(t_1 \cdots t_n) = \prod_{i=1}^{n} P(t_i | t_{i-1}, t_{i-2})$$

with appropriate treatment of boundaries. The trigram probabilities are smoothed as for the HMM tagger. The combined model is relatively insensitive to the relative weights of the two component models, as specified by λ; in any case, exact optimization of this parameter was not performed. We refer to this model as Combined. The Combined model is not a sound probabilistic model as it does not define a probability distribution over parse trees. It does however provide a crude way to combine ancestor and left context information.

The second column in Table 2 shows the accuracy of parse selection using the models described above. For comparison, a baseline showing the expected performance of choosing parses randomly according to a uniform distribution is included as the first row. The accuracy results are averaged over a ten-fold cross-validation on the data set summarized in Table 1. The data we used for this experiment was the set of disambiguated sentences that have exactly one preferred parse (comprising a total of 5312 sentences). Often the stochastic models we are considering give the same score to several different parses. When a model ranks a set of m parses highest with equal scores and one of those parses is the preferred parse in the treebank, we compute the accuracy on this sentence as $1/m$.

Since our approach of defining the probability of analyses using derivation trees is different from the traditional approach of learning PCFG grammars from phrase structure trees, a comparison of the two is probably in order. We tested the model PCFG-GP defined over the corresponding phrase structure trees and its average accuracy was 65.65% which is much lower than the accuracy of the same model over derivation trees (71.73%). This result suggests that the information about grammar constructions is very helpful for parse disambiguation.

Method		**Task**	
		tag sel.	parse sel.
Random		90.13%	25.81%
Tagger	unigram	96.75%	44.15%
	trigram	97.87%	47.74%
	oracle	100.00%	54.59%
PCFG	simple	97.40%	66.26%
	grandparent	97.43%	71.73%
	combined	98.08%	74.03%

Table 2: Performance of the HMM and PCFG models for the tag and parse selection tasks (accuracy).

The results in Table 2 indicate that high disambiguation accuracy can be achieved using very simple statistical models. The performance of the perfect tagger shows that, informally speaking, roughly half of the information necessary to disambiguate parses is available in the lexical types alone. About half of the remaining information is recovered by our best method, Combined.

An alternative (more primitive) task is the tagging task itself. It is interesting to know how much the tagging task can be improved by perfecting parse disambiguation. With the availability of a parser, we can examine the accuracy of the tag sequence of the highest scoring parse, rather than trying to tag the word sequence directly. We refer to this problem as the *tag selection* problem, by analogy with the relation between the parsing problem and the parse selection problem. The first column of Table 2 presents the performance of the models on the tag selection problem. The results are averaged accuracies over 10 cross-validation splits of the same corpus as the previous experiment, and show that parse disambiguation using information beyond the lexical type sequence slightly improves tag selection performance. Note that in these experiments, the models are used to rank the tag sequences of the possible parses and not to find the most probable tag sequence. Therefore tagging accuracy results are higher than they would be in the latter case.

Since our corpus has relatively short sentences and low ambiguity it is interesting to see how much the performance degrades as we move to longer and more highly ambiguous sentences. For this purpose, we report in Table 3 the parse ranking accuracy of the Combined model as a function of the number of possible analyses for sentences. Each row corresponds to a set of sentences with number of possible analyses greater or equal to the bound shown in the first column. For example, the first row contains information for the sentences with ambiguity ≥ 2, which is all ambiguous sentences. The columns show the total number of sentences in the set, the expected accuracy of guessing at random, and the accuracy of the Combined model. We can see that the parse ranking accuracy is decreasing quickly and more powerful models will be needed to achieve good accuracy for highly ambiguous sentences.

Despite several differences in corpus size and compo-

Analyses	Sentences	Random	Combined
≥ 2	3824	25.81%	74.03%
≥ 5	1789	9.66%	59.64%
≥ 10	1027	5.33%	51.61%
≥ 20	525	3.03%	45.33%

Table 3: Parse ranking accuracy by number of possible parses.

sition, it is perhaps nevertheless useful to compare this work with other work on parse selection for unification-based grammars. Johnson et al. (1999) estimate a Stochastic Unification Based Grammar (SUBG) using a log-linear model. The features they include in the model are not limited to production rule features but also adjunct and argument and other linguistically motivated features. On a dataset of 540 sentences (total training and test set) from a Verbmobil corpus they report parse disambiguation accuracy of 58.7% given a baseline accuracy for choosing at random of 9.7%. The random baseline is much lower than ours for the full data set, but it is comparable for the random baseline for sentences with more than 5 analyses. The accuracy of our Combined model for these sentences is 59.64%, so the accuracies of the two models seem fairly similar.

4 Related Work

To the best of our knowledge, no prior research has been conducted exploring the linguistic depth, flexibility in available information, and dynamic nature of treebanks that we have proposed. Earlier work on building corpora of hand-selected analyses relative to an existing broad-coverage grammar was carried out at Xerox PARC, SRI Cambridge, and Microsoft Research. As all these resources are tuned to proprietary grammars and analysis engines, the resulting treebanks are not publicly available, nor have reported research results been reproducible. Yet, especially in light of the successful LinGO open-source repository, it seems vital that both the treebank and associated processing schemes and stochastic models be available to the general (academic) public. An on-going initiative at Rijksuniversiteit Groningen (NL) is developing a treebank of dependency structures (Mullen, Malouf, & Noord, 2001), derived from an HPSG-like grammar of Dutch (Bouma, Noord, & Malouf, 2001). The general approach resembles the Redwoods initiative (specifically the discriminator-based method of tree selection; the LKB tree comparison tool was originally developed by Malouf, after all), but it provides only a single stratum of representation, and has no provision for evolving analyses in tandem with the grammar. Dipper (2000) presents the application of a broad-coverage LFG grammar for German to constructing tectogrammatical structures for the TiGer corpus. The approach is similar to the Groningen framework, and shares its limitations.

References

Abney, S. P. (1997). Stochastic attribute-value grammars. *Computational Linguistics*, *23*, 597 – 618.

Agresti, A. (1990). *Categorical data analysis.* John Wiley & Sons.

Bouma, G., Noord, G. van, & Malouf, R. (2001). Alpino. Wide-coverage computational analysis of Dutch. In W. Daelemans, K. Sima-an, J. Veenstra, & J. Zavrel (Eds.), *Computational linguistics in the Netherlands* (pp. 45 – 59). Amsterdam, The Netherlands: Rodopi.

Carter, D. (1997). The TreeBanker. A tool for supervised training of parsed corpora. In *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering.* Madrid, Spain.

Charniak, E. (1997). Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence* (pp. 598 – 603). Providence, RI.

Charniak, E., & Caroll, G. (1994). Context-sensitive statistics for improved grammatical language models. In *Proceedings of the Twelth National Conference on Artificial Intelligence* (pp. 742 – 747). Seattle, WA.

Collins, M. J. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Meeting of the Association for Computational Linguistics and the 7th Conference of the European Chapter of the ACL* (pp. 16 – 23). Madrid, Spain.

Copestake, A. (2002). *Implementing typed feature structure grammars.* Stanford, CA: CSLI Publications.

Copestake, A., Lascarides, A., & Flickinger, D. (2001). An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Meeting of the Association for Computational Linguistics.* Toulouse, France.

Dipper, S. (2000). Grammar-based corpus annotation. In *Workshop on linguistically interpreted corpora LINC-2000* (pp. 56 – 64). Luxembourg.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering*, *6 (1) (Special Issue on Efficient Processing with HPSG)*, 15 – 28.

Harris, T. E. (1963). *The theory of branching processes.* Berlin, Germany: Springer.

Johnson, M., Geman, S., Canon, S., Chi, Z., & Riezler, S. (1999). Estimators for stochastic ‘unification-based’ grammars. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics* (pp. 535 – 541). College Park, MD.

Manning, C. D., & Schütze, H. (1999). *Foundations of statistical Natural Language Processing.* Cambridge, MA: MIT Press.

Mullen, T., Malouf, R., & Noord, G. van. (2001). Statistical parsing of Dutch using Maximum Entropy models with feature merging. In *Proceedings of the Natural Language Processing Pacific Rim Symposium.* Tokyo, Japan.

Oepen, S., & Callmeier, U. (2000). Measure for measure: Parser cross-fertilization. Towards increased component comparability and exchange. In *Proceedings of the 6th International Workshop on Parsing Technologies* (pp. 183 – 194). Trento, Italy.

Toutanova, K., & Manning, C. D. (2002). Feature selection for a rich HPSG grammar using decision trees. In *Proceedings of the sixth conference on natural language learning (CoNLL-2002).* Taipei.

Wahlster, W. (Ed.). (2000). *Verbmobil. Foundations of speech-to-speech translation.* Berlin, Germany: Springer.