# Towards Holistic Grammar Engineering and Testing

— Grafting Treebank Maintenance into the Grammar Revision Cycle —

**Stephan Oepen**
Universitetet i Oslo
& CSLI Stanford
oe@csli.stanford.edu

**Dan Flickinger**
CSLI Stanford
dan@csli.stanford.edu

**Francis Bond**
NTT Communication
Science Laboratories
bond@cslab.kecl.ntt.co.jp

## Abstract

We present a new methodology for the semi-automated maintenance of a treebank built from analyses of a computational grammar and gauge the effort required for each update cycle. Based on a decade of large-scale grammar engineering experience, we propose a tight integration of treebank maintenance with the continuous evolution of a 'deep' computational grammar.

## 1 Background & Motivation

Moving (on) into the new millennium, it has become common-place folklore in our field to accept the complementary nature of linguistic ('symbolic') and stochastic ('data-driven') approaches to NLP. A majority of NLP taks and applications, today, requires the combination of both research strains, and, accordingly, linguistic description and machine learning are no longer viewed as competing paradigms. At the same time, there is an emerging demand for 'richer' annotation of training corpora, specifically treebanks that include more than coarse-grained phrase structure information. With few exceptions (notably the Prague Dependency Bank; Hajic, 1998), however, work on more closely relating the actual annotations in treebanks to contemporary linguistic research is largely lacking.

The LinGO Redwoods Treebank (Oepen et al., 2002) is a treebank comprised entirely of analyses derived from a broad-coverage computational grammar, the LinGO English Resource Grammar (ERG; Flickinger, 2000). The ERG is a large-scale HPSG implementation, actively developed at Stanford since 1993, and its analyses provide precise, fine-grained syntactic and semantic information; Minimal Recursion Semantics (MRS) is the general framework for meaning representation. Building on an array of existing software tools for processing with the ERG (and similar grammars), the Redwoods Treebank was constructed by parsing selected domain corpora and subsequently hand-inspecting analyses and selecting the intended reading(s) for each input item. Annotation (i.e. manual parse selection) in Redwoods builds on the notion of *elementary discriminants* (Carter, 1997), basic properties of sub-constituents in the parse forest that account for contrasts (i.e. local sources of ambiguity) among analyses. Discriminants—competing lexical entries, for example, or a choice of using the head – complement vs. head – adjunct schema to build a token phrase—are fairly easy to judge, even for non-experts, and enable annotators to navigate the parse forest quickly. Using a specialized tool, each annotator decision on accepting or rejecting a discriminant results in the elimination of large parts of the parse forest, so that a small number of local decisions typically will be sufficient to disambiguate even highly ambiguous inputs. Table 1 summarizes the Redwoods development status to date.[1]

While the general Redwoods philosophy is well-documented, in the following we motivate a tight integration of treebank maintenance and grammar regression testing (Section 2) and present a novel methodology for mostly-automated treebank updates (Section 3); empirical results from two update cycles and subsequent grammar development experience strongly suggest that the proposed coupling of grammar engineering and corpus maintenance offer a new quality of revision engineering, with mutual benefits to both tasks.

## 2 Treebanks in Grammar Engineering

The challenges in large-scale grammar development are many-fold and multi-dimensional, and rigorous and systematic regression testing on structured test

[1] Up to now, the treebank was mainly used for training and evaluation of stochastic parse selection techniques (Toutanova & Manning, 2002).

corpora was found central to our empiricist methodology dubbed *competence and performance profiling* (Oepen & Flickinger, 1998); subsequently, we demonstrated how precise, near-instantaneous feedback on grammar revisions and fine-grained record keeping can enable truly parallel development, i.e. multiple grammarians at remote sites contributing to the same resource on a daily basis (Oepen, Bender, Callmeier, Flickinger, & Siegel, 2002).

In more recent work, we have found that the availability of Redwoods-type treebanks offers a new grade of 'holistic' regression testing. Knowledge about the intended analysis at various levels (e.g. as a specific derivation, labeled parse tree, or MRS semantics) facilitates focussed comparison of results obtained from a grammar revision to earlier records. Where additional analyses have been added, for example, we can still confirm that the right derivation is among them; where the derivation itself may have changed, the labelled tree or MRS may still be equivalent; reading the MRS off the intended parse enables targeted batch testing of the generation component and, among others, confirmation that the original string is among the paraphrases. In summary, testing against a disambiguated parsed corpus has become a central component in grammar development, and each new grammar release is now accompanied by an update of (substantial parts of) the corresponding treebank.

## 3 Treebank Maintenance

Among the more challenging aspects of our Redwoods research was the search for a methodology for automated updates of the treebank, to keep track with the continuous evolution of the underlying linguistic framework. We have arrived at an innovative procedure that—crucially building on the notion of elementary linguistic discriminants—allows us to maintain the treebank with minimal manual effort. In fact, our semi-automatic update procedure directly helps grammarians in identifying and isolating effects of changes made in the grammar.

Generally speaking, the update procedure attempts to carry forward the disambiguating decisions made by annotators from one (older) version of the base corpus to a newer version (obtained by re-parsing the data with a revised grammar). As annotator decisions on elementary discriminants disambiguate (often) isolated local regions of alternation—and do so by virtue of (mostly) independent syntacto-semantic properties—even in the presence of major changes in the grammar at least part of the disambiguating decisions should be reusable. Furthermore, whenever annotators toggle a discriminant, the software determines the set of decisions entailed by the decision just made, i.e. negative discriminants that are incompatible with the remaining set of active parses or positive discriminants that are known to be equivalent to the one just toggled. Both types of decisions are recorded at annotation time and—in conjunction with the (desirable) redundancy already present in the use of partly overlapping discriminants—make the record keeping of 'disambiguating potential' highly redundant.

Table 1: LinGO Redwoods Development status. Data from three sources has been annotated systematically, viz. VerbMobil dialogues, ecommerce customer email, and excerpts from tourism brochures; sections highlighted in italics are maintained through each grammar revision, together with a fourth data set (the TREC-8 questions), although it is not formally part of Redwoods. The columns are, from left to right, the number of sentences ('♯'), average length ('‖'), and structural ambiguity ('×'), broken down for three subsets, viz. items (i) for which annotators rejected all analyses derived from the grammar (no active trees), (ii) where annotation resulted in exactly one preferred analysis (one active tree), and (iii) where full disambiguation was not accomplished (more than one active tree).

| | | **active = 0** | | | **active = 1** | | | **active > 1** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | ♯ | ‖ | × | ♯ | ‖ | × | ♯ | ‖ | × |
| VerbMobil | ***VM6*** | *15* | *14·3* | *8670* | *3811* | *7·9* | *111* | *0* | *0·0* | *0* |
| | **VM13** | 248 | 10·8 | 80 | 2028 | 8·7 | 59 | 3 | 15·5 | 198 |
| | **VM31** | 216 | 10·1 | 95 | 1746 | 7·5 | 30 | 5 | 8·4 | 20 |
| | **VM32** | 16 | 11·8 | 57 | 681 | 8·4 | 53 | 0 | 0·0 | 0 |
| ecommerce | **ECPA** | 156 | 10·2 | 19 | 1026 | 8·2 | 12 | 9 | 8·2 | 13 |
| | **ECOS** | 144 | 12·5 | 143 | 1088 | 8·0 | 18 | 24 | 11·6 | 37 |
| | **ECPR** | 81 | 11·9 | 46 | 899 | 7·4 | 11 | 5 | 10·4 | 49 |
| | ***ECOC*** | *38* | *13·1* | *259* | *1144* | *7·4* | *47* | *2* | *6·0* | *21* |
| | ***TREC*** | *4* | *11·5* | 86 | *662* | *7·9* | *20* | *0* | *0·0* | *0* |
| | ***HIKE*** | *1* | *22·0* | *876* | *318* | *12·9* | *187* | *0* | *0·0* | *0* |
| | **Total** | **919** | **11·1** | **229** | **13403** | **8·1** | **60** | **48** | **10·5** | **42** |

A full, semi-automated update cycle for the Redwoods treebank proceeds along the following steps:

(1) *corpus preparation* using the new grammar, obtain a new 'target' corpus by running the parser on it and recording all derivations in the [incr tsdb()] database;

(2) *automated update* for each item in the new corpus, extract the set of discriminants and in-

tersect it with recorded decisions for this sentence in the earlier corpus;

(3) *manual resolution* a user-supplied predicate decides, for each item, whether the update was successful and complete; remaining items require subsequent annotator inspection.

Although for a grammar like the ERG 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 (i) that one set of discriminants will always fully disambiguate a more recent set of analyses for the same utterance (as the grammar may introduce additional distinctions, i.e. more ambiguity), (ii) nor that all recorded discriminants will have a matching property in the new corpus (i.e. where the grammar has recast or simply collapsed distinctions), (iii) nor that (seemingly) successfully re-playing a history of disambiguating decisions will necessarily identify the correct, preferred analysis for all sentences. While the third observation suggests that, in principle, one might arrive at a dis-preferred parse even when all recorded discriminants match the new corpus and yield the expected number of active parses (typically one), this is of no concern in practice: a grammarian would have to deliberately rename and systematically swap elementary properties to achieve such an effect. Likewise, the second source of potential mismatches in the update cycle (viz. item (ii) from our list) is mitigated to a certain extent through the overlap (redundancy) in the recorded decisions. Finally, the first concern (item (i) above) directly relates to information that should usually be highly relevant to the grammar writer when assessing the impact of recent changes made to the grammar.

To gauge the practical feasibility of our update procedure, we analyzed records obtained during the first semi-automated Redwoods update cycle (resulting in the 3rd Growth version). For this exercise to be a strong measure of how much disambiguating information can be retained across grammar changes, we let close to eighteen months pass before attempting the first update. Between June 2001 and October 2002, the ERG was actively used in building a commercial product (for automated email response) and adapted from the original VerbMobil (spoken dialogue) domain to ecommerce customer emails. Accordingly, the 'distance' between the two versions of the grammar used in the treebank update reported here is exceptionally large. Table 2 compiles a summary of changes made to the grammar between June 2001 and October 2002.[2] Between the two ERG versions, differentials range between fourteen and forty five per cent for some central measures. Clearly, the scope of the update problem is much bigger in this scenario than would usually be expected.

Table 2: Quantitative assessment of evolution between the June 2001 and October 2002 versions of the ERG. The column labeled $\Delta$ indicates the differential of change, where two values indicate that part of the original was eliminated while, at the same time, new objects were added. The apparently stable absolute numbers of appropriate features, for example, are misleading in that the two sets only intersect in 137 elements, i.e. nine original features were replaced by ten new features.

| | **jun-01** | **oct-02** | $\Delta$ |
|---|---|---|---|
| **distinct features** | 148 | 149 | $-6\%$ $+7\%$ |
| **type hierarchy** | 3,062 | 3,895 | $+27\%$ |
| **grammar rules** | 86 | 94 | $-11\%$ $+26\%$ |
| **lexical types** | 400 | 580 | $+45\%$ |
| **semantic relations** | 5,406 | 6,162 | $+14\%$ |
| **lexical entries** | 8,135 | 9,954 | $+22\%$ |
| **lines of source** | 25,847 | 32,199 | $+25\%$ |

Practical update results are summarized in Table 3, showing a range of relevant measures. The update procedure itself provided immediate feedback to the grammarian that resulted in a series of three engineering cycles iterating the update procedure and further revisions to the grammar as a response to observations made during the update cycle; this micro-level experimentation was carried out on two of the four VerbMobil dialogues, while the remaining two were only updated once the grammarian had converged on the final version of the ERG for the 3rd Growth treebank. The direct transition from the June 2001 to the October 2002 version is depicted in the upper half of Table 3 and shows that close to sixty per cent of the (ambiguous) sentences in the corpus required no manual intervention, i.e. no additional annotator decisions to fully disambiguate the parse forest after the application of recorded discriminants from the earlier corpus. This surprising result comes despite the fact that roughly half of the discriminants

[2] Although it is in general hard to quantify grammar evolution and compare across grammar versions, some of the measures reported in Table 2 immediately pertain to the type of information used in Redwoods discriminants.

Table 3: Quantitative summary of semi-automated update, considering ambiguous items only: the table reflects the amount of manual intervention for two distinct update scenarios, viz. one update after eighteen months of grammar evolution and a second after three weeks (labeled '$VM_{13+31}$' and '$VM_{6+32}$', respectively). Each data set is aggregated by the number of manual decisions (the parameter *new* recorded by the software) required in the update for full disambiguation of the new corpus, where 'new = 0' indicates a fully-automated update. The columns are, from left to right, the total number of items in each aggregate, average number of active ('in') and rejected ('out') parses in the original corpus, average number of discriminants that were successfully carried over ('yes') or had to be discarded ('no')', in and out parses in the new corpus after applying the discrimants, average number of additional (manual) annotator decisions, and the ultimate number of in and out parses.

| Aggregate | | items | original | | matches | | update | | | final | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | in | out | yes | no | in | out | new | in | out |
| | | ♯ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ | $\phi$ |
| $VM_{13+31}$ | **new = 0** | 1421 | 1·1 | 23·6 | 8·1 | 8·5 | 1·0 | 13·9 | 0·0 | 1·0 | 13·9 |
| | **new = 1** | 708 | 1·1 | 38·1 | 6·9 | 9·8 | 2·2 | 29·6 | 1·0 | 1·0 | 30·8 |
| | **new ≥ 2** | 273 | 1·3 | 61·5 | 12·1 | 15·2 | 4·2 | 72·0 | 2·8 | 1·0 | 75·2 |
| | **Total** | **2402** | **1·1** | **32·2** | **8·2** | **9·6** | **1·8** | **25·1** | **0·6** | **1·0** | **25·9** |
| $VM_{6+32}$ | **new = 0** | 2195 | 1·0 | 72·2 | 17·2 | 1·0 | 1·0 | 69·3 | 0·0 | 1·0 | 69·3 |
| | **new = 1** | 73 | 1·0 | 31·9 | 11·7 | 1·4 | 2·2 | 116·0 | 1·0 | 1·0 | 117·3 |
| | **new ≥ 2** | 20 | 1·0 | 192·6 | 13·3 | 0·8 | 16·7 | 297·5 | 2·9 | 1·0 | 313·2 |
| | **Total** | **2288** | **1·0** | **72·0** | **17·0** | **1·1** | **1·2** | **72·8** | **0·1** | **1·0** | **73·0** |

had to be discarded during the update because they no longer had a corresponding property in the target parse forest. For the remainder of the data set a slightly smaller percentage of the recorded decisions could be re-used (for an overall average re-use ratio of forty six per cent), but still the vast majority of items, on average, did not require more than a single additional decision from annotators to achieve complete disambiguation. This appears to, in part, be due to a stable average ambiguity rate across the two data sets even though—given revisions in the grammar—no two derivations would yield an exact match. The lower part of Table 3, finally, seems to confirm the power of our discriminant-based update procedure in that—this time across two grammar versions that are only three weeks apart from each other—the full cycle on 2,288 ambiguous items required a total of one hundred and thirty additional annotator decisions.

## 4 Conclusion & Outlook

Given the full integration of the update procedure and annotation environment, a full treebank update (across reasonably similar grammar versions) can be completed in a matter of minutes or hours and, hence, is now part of the standard grammar regression testing and release cycle. We have found that the integration of treebank maintenance and grammar engineering is immediately beneficial to both tasks. A similar observation holds for experimentation with stochastic models trained on the Redwoods data—thereby adding another dimension to our holistic testing scenario—which we plan to document in comparable detail in the near future.

## References


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.

Flickinger, D. (2000). On building a more efficient grammar by exploiting types. *Natural Language Engineering, 6 (1)*, 15 – 28.

Hajic, J. (1998). Building a syntactically annotated corpus. the Prague dependency treebank. In *Issues of valency and meaning* (pp. 106 – 132). Prague, Czech Republic: Karolinum.

Oepen, S., Bender, E. M., Callmeier, U., Flickinger, D., & Siegel, M. (2002). Parallel distributed grammar engineering for practical applications. In *Proceedings of the Workshop on Grammar Engineering and Evaluation.* Taipei, Taiwan.

Oepen, S., & Flickinger, D. (1998). Towards systematic grammar profiling. Test suite technology ten years after. *Journal of Computer Speech and Language, 12 (4)*, 411 – 436.

Oepen, S., Toutanova, K., Shieber, S., Manning, C., Flickinger, D., & Brants, T. (2002). The LinGO Redwoods Treebank. Motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics.* Taipei, Taiwan.

Toutanova, K., & Manning, C. D. (2002). Feature selection for a rich HPSG grammar using decision trees. In *Proceedings of the 6th Conference on Natural Language Learning.* Taipei, Taiwan.