Research on Language and Computation (2004) 2: 575–596
DOI 10.1007/s11168-004-7430-4

LinGO Redwoods

A Rich and Dynamic Treebank for HPSG

STEPHAN OEPEN[1], DAN FLICKINGER[1], KRISTINA TOUTANOVA[2], and CHRISTOPHER D. MANNING[2]
[1]*Center for the Study of Language and Information, Stanford University, Ventura Hall, Stanford, CA 94305, USA (E-mail: oe, dan@csli.stanford.edu);* [2]*Department of Computer Science, Stanford University, Stanford, CA 94305, USA (E-mail: kristina, manning@cs.stanford.edu)*

Abstract. Reflecting an increased need for stochastic parse selection models over hand-built linguistic grammars and a lack of appropriately detailed training material, we present the Linguistic Grammars On-Line (LinGO) Redwoods initiative, a seed activity in the design and development of a new type of treebank. LinGO Redwoods aims at the development of a novel treebanking methodology, (i) *rich* in nature and *dynamic* in both (ii) the ways linguistic data can be retrieved from the treebank in varying granularity and (iii) the constant evolution and regular updating of the treebank itself, synchronized to the development of ideas in syntactic theory. Starting in June 2001, the project has been working to build the foundations for this new type of treebank, develop a basic set of tools required for treebank construction and maintenance, and construct an initial set of 10,000 annotated trees to be distributed together with the tools under an open-source license.

Key words: HPSG, parse selection, treebank maintenance, treebanks

1. Background

From machine translation to speech recognition and information extraction and retrieval engines, a wide range of applications demand increasing accuracy and robustness from natural language processing. Meeting these demands for precise linguistic analysis will require hand-built, in-depth grammars of natural language. Among others, Head-Driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994) has been one of the predominant paradigms for building such grammars. The Linguistic Grammars On-Line (LinGO) Project at CSLI has been conducting research and development in HPSG implementation since 1994. Jointly with international partners – primarily at Saarbrücken (Germany), Cambridge, Edinburgh, and Sussex (UK), and Tokyo (Japan) – the LinGO initiative has developed a broad-coverage, precise HPSG implementation of English (the LinGO English Resource Grammar, ERG; Flickinger, 2000), a framework for

semantic composition in large-scale computational grammars (Minimal Recursion Semantics, MRS; Copestake et al., 1999, 2001), and an advanced grammar development environment (the LKB system; Copestake, 1992, 2002; Malouf et al., 2002). Through contributions from collaborating partners, a pool of open-source HPSG resources has developed that now includes broad-coverage grammars for several languages, a common profiling and benchmarking environment (Oepen and Callmeier, 2000), and an industrial-strength C^{++} run-time engine for HPSG grammars (Callmeier, 2000). LinGO resources are in use world-wide for teaching, research, and application building. Because of their wide distribution and common acceptance, the HPSG framework and LinGO resources provide a good anchor point for the Redwoods treebanking initiative introduced here.[1]

2. Combining Linguistic and Stochastic Approaches

For the past decade or more, symbolic, linguistically-oriented methods (like those pursued within the HPSG framework) and statistical or machine learning approaches to NLP have typically been perceived as incompatible or even competing paradigms; the former, more traditional approaches are often referred to as 'deep' NLP, in contrast to the comparatively recent branch of language technology focusing on 'shallow' (text) processing methods. Shallow processing techniques have produced useful results in many classes of applications, but 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 HPSG-type systems in certain types of real-world applications, and fully-automated, deep grammatical analysis of unrestricted text remains an unresolved challenge.

In particular, applications of analytical grammars for natural language parsing or generation require the use of sophisticated statistical techniques for resolving ambiguities. While precise linguistic grammars tend to assign a relatively small number of analyses to an average-length string (typically dozens, sometimes hundreds, rarely thousands) and fare well in rejecting ungrammatical input, attempts at encoding preferences of grammatical analyses manually have failed in practice. One common technique, the inclusion of sortal constraints on arguments of semantic relations, e.g. animacy on subjects of agentive predicates (Müller and Kasper, 2000), risks contaminating the grammar with non-linguistic knowledge about real-world regularities or properties of a specific domain; and computers and projects die too, after all. Another traditional approach, annotating the core grammar with heuristic preferences or hand-constructed measures of 'likelihood'

for a given domain (see, for example, Kiefer et al., 1999; King et al., 2000), has been found severely limited in its scalability, very time-consuming and error-prone, and impossible to port across domains. In short, grammarians should not have to express a pseudo-analytical preference between the strict transitive or ditransitive uses of *sell* in examples like the following:

(1) *Do you sell IBM laptops?*
(2) *Do you sell IBM ice cream?*

Although resolving ambiguities of this type generally would seem to require world and domain knowledge and suitable inference capabilities, it is to be expected that a stochastic model of, say, word to word relations alone will be able to approximate the solution well.

We observe general consensus on the necessity for bridging activities, combining symbolic and stochastic approaches to NLP. At the same time, the transfer of HPSG resources into industry – where a typical application will expect to consume only one or a small number of results from linguistic analysis – has amplified the need for general parse ranking, disambiguation, and robust recovery techniques which all require suitable stochastic models for HPSG processing. While there is active research in stochastic parsing in a number of frameworks, HPSG still exhibits a lack of appropriately rich and dynamic language corpora. Likewise, stochastic parsing has so far been focused on IE-type applications and typically lacks any depth of semantic interpretation. The Redwoods initiative is designed to fill in this gap.

3. Why Another (Type of) Treebank?

Most probabilistic parsing research – including, for example, work in the tradition of Collins (1997) and Charniak (1997) – is based on branching process models (Harris, 1963). An important recent advance in this area has been the application of log-linear models (Agresti, 1990) to modeling linguistic systems. These models can deal with the many interacting dependencies and the immense structural complexity of constraint-based or unification-based theories of syntax (Johnson et al., 1999).

While several medium- to large-scale treebanks exist for English (and some for other major languages), pre-existing publicly available resources – as for example the widely recognized Penn Treebank (PTB; Marcus et al., 1993), the German TiGer Corpus (Skut et al., 1997), the Prague Dependency Treebank (Hajic, 1998), or the Dutch Alpino Dependency Bank (van der Beek et al., 2002) – exhibit the following limitations: (i) each resource has chosen to focus on a single stratum of linguistic description, either topological (phrase structure) or tectogrammatical (dependency structure), (ii) the depth of linguistic information recorded in these treebanks is comparatively shallow (limited syntax, little or no semantics), (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 decade-long) evolution of a large-scale treebank tend to fall behind advances in formal linguistics and grammatical representation.

Conversely, a hand-built precision grammar like the LinGO ERG encodes linguistic distinctions at various descriptive levels and in a granularity much finer than is found in existing annotated corpora. A treebank like the PTB, for example, failing to distinguish syntactic arguments from 'free' modifiers, will not provide sufficient detail to resolve complement–adjunct ambiguities (e.g. in *Pack your suitcase in the car!* where the prepositional phrase can either be the directional target or general location of the packing activity) which are analyzed in the ERG. Likewise, limiting linguistic description to dependency information only potentially blurs the structural difference in subject–object ambiguities commonly exhibited by verb-second languages: the Norwegian utterance *Kari ser Gyrd.* ('Kari sees Gyrd.'), for example, has a second analysis as a topicalized structure (with Gyrd doing the seeing), but topicalization, presumably, is not reflected at the tectogrammatical level; hence, looking at dependency structures only, it would be impossible for stochastic models to acquire a preference for non-topicalized structures.

Research on the definition and acquisition of stochastic models that can be used in conjunction with broad-coverage HPSG grammars like the LinGO ERG requires annotated corpora that provide an adequate match of available information and linguistic granularity to the grammars. The availability of even a medium-sized treebank would allow us to begin exploring the use of these models for probabilistic disambiguation of HPSG grammars. At the same time, other researchers have started work on stochastic HPSG (or are about to), some pursuing unsupervised approaches, but in many cases using the same grammar or at least the same descriptive formalism and grammar engineering environment. The availability of a reasonably large, hand-disambiguated HPSG treebank is expected to greatly facilitate comparability of results and models obtained by various groups and, eventually, to help define a common evaluation metric.

4. LinGO Redwoods – A Rich and Dynamic Treebank

In response to the demand for stochastic parse selection models over HPSG grammars and the lack of suitable training material, the LinGO Laboratory at CSLI has started work on a novel type of treebank for English, dubbed LinGO Redwoods.[2] Some important innovative aspects of the Redwoods approach to treebanking are (i) its anchoring of all linguistic data captured in the treebank in the HPSG framework and the publicly available

LinGO English Resource Grammar, (ii) the organization of the annotation process and subsequent update of the treebank around elementary linguistic properties (called *discriminant*, see Section 4.1 below), and (iii) its provision of tools for the extraction of various user-defined representations. Unlike in 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 provides tools to extract many different types of linguistic information at greatly varying granularity. In this respect, the Redwoods treebank is *rich* in linguistic information and *dynamic* in both how the content is presented to users and how it is maintained over time.

4.1. The source of ambiguity: basic discriminants

The Redwoods annotation environment is configured from two pre-existing pieces of software, viz. (i) a tree comparison tool (similar in kind to the SRI Cambridge TreeBanker; Carter, 1997) that is part of the LKB grammar development system and (ii) the [incr tsdb()] profiling environment (essentially a specialized database recording fine-grained parsing results obtained from a HPSG system; Oepen and Callmeier, 2000a). Thus, the treebank is constructed as an extension of the existing [incr tsdb()] data model and tools, providing annotators with a way of selecting the preferred analysis for a string efficiently and recording the resulting preference and all decisions made in the database.

The tree comparison tool presents annotators, one sentence at a time, with the full set of analyses produced by the grammar together with a condensed view of where the ambiguity, lexical and phrasal, that gives rise to this set of analyses originates. Put simply, the full set of analyses reflects the cross product of a series of more local choices – alternation between lexical entries or alternatives for modifier attachment, for example – of which some are independent of each other while others may mutually interact. The tool extracts elementary linguistic properties – called *discriminants* – that correspond to local ambiguity and uses the inference rules of Carter (1997) to determine the smallest possible set of discriminants that fully disambiguates the parse forest. When presented with individual local properties as they indicate choice points in assigning the linguistic analysis to the token sentence, annotators can quickly navigate through the parse forest and identify the correct or preferred analysis in the current context (or, in rare cases, end up rejecting all analyses proposed by the grammar). Using the discriminant-based approach to tree comparison, and given the elementary nature of each decision, annotators need little expert knowledge of the underlying grammar, but instead decide on a range of properties that distinguish competing analyses and are relatively easy to judge.

For each discriminant, annotators can choose whether they require the indicated property in the intended analysis (i.e. positively select a discriminant) or disallow it (i.e. negatively reject a discriminant). Each annotator decision reduces the set of active analyses – trimming down the parse forest – as for positive decisions only trees that have the indicated property remain available, whereas with negative decisions all trees with the rejected property can be excluded. As the set of active analyses is incrementally reduced, so is the set of discriminants: discriminants from the original set that either have no remaining active parse or are compatible with all remaining parses can be suppressed from the annotator display, as deciding on these properties will not further disambiguate the parse forest.

While the general Redwoods approach makes no implicit commitment as to the exact nature of discriminants, it is important to maintain a fine balance between, on the one hand, sufficient information for effective and full disambiguation and, on the other hand, locality and simplicity of individual decisions.[3]

While working with the LinGO English Resource Grammar, we are using the following four types of elementary properties, of which the first two only apply to phrases and words, respectively, while the latter two can be extracted from either a lexical or non-lexical constituent.

- *constituents* use of a particular construction (i.e. a rule of the grammar) over a specific substring of the input;
- *lexical items* use of a particular lexical entry (identified by its lexical type or 'part of speech') for a specific input token;
- *semantics* appearance of a particular key relation (primary predicate) on a specific constituent; and
- *labeling* assignment of a particular abbreviatory phrase structure node label to a specific constituent.

Obviously, this set of discriminants already creates potential for redundancy as, for example, lexical alternation will often be reflected at both the lexical type and semantic levels, so that annotators may identify the intended lexical item through two (apparently) independent discriminants. Again, striking the right balance between expository parsimony and informational benefit provided to annotators is an empirical problem; the Redwoods tools offer a number of switches to selectively enable or exclude redundant types of discriminants, so that expert annotators can choose to operate on a maximally concise set of decisions while novice annotators can allow themselves additional discriminants which they may be more comfortable deciding on. However, in either mode our discriminant-based disambiguation approach is different from the context variables associated to packed ambiguity in LFG f-structure (as implemented by the Xerox Linguistic Environment; King et al., 2000), since a packed f-structure can only present a reasonably compact summary of *all* information in the

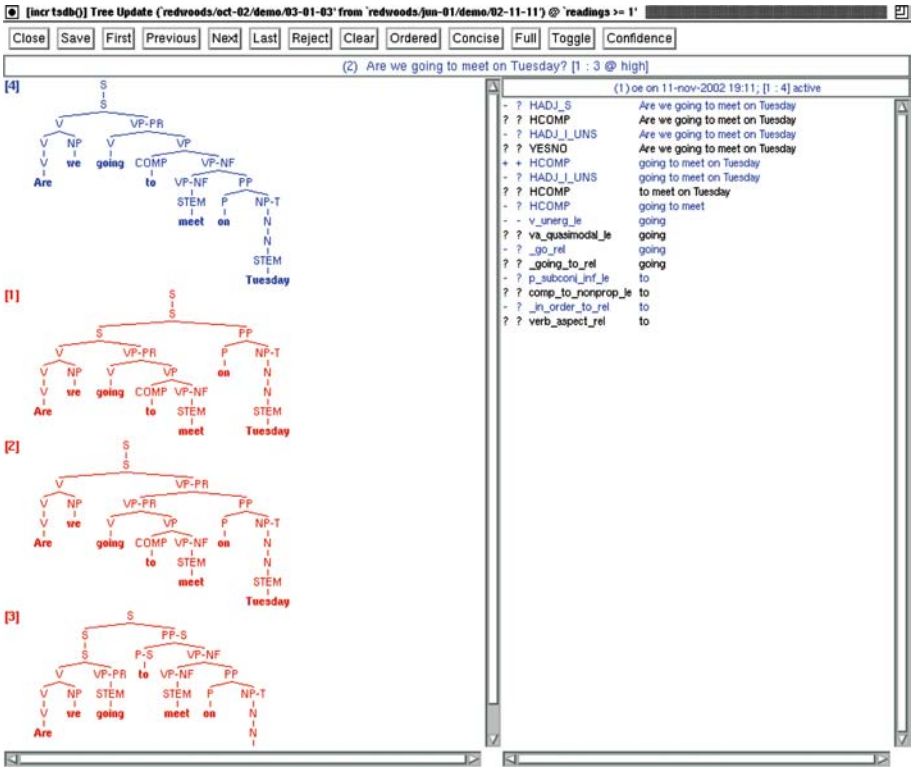

Figure 1. Screenshot of Redwoods treebanking environment: the window on the left presents the full set of analyses as labeled phrase structure trees (often too numerous to fit on a single page), and the window on the right shows the minimal set of discriminating properties, based on either a particular lexical item (with an '_le' suffix), semantic relation (a '_rel' suffix), or syntactic construction (in all capitals) applied to a specific substring to form a constituent.

parse forest while a set of elementary discriminants reduces the information presented to annotators to the basic amount of structure required to completely disambiguate the input sentence.

Figure 1 presents the Redwoods annotation environment. For a second-year Stanford undergraduate in linguistics, our approach to parse selection through minimal discriminators turned out to be not at all hard to learn and required less training in specifics of the grammatical analyses delivered by the LinGO grammar than could have been expected. After 3–4 weeks in hands-on training, the annotator was able to disambiguate at a rate of about 2000 sentences per week; annotator throughput is enhanced by the ability of the treebanking environment to only partially disambiguate a sentence and flag it for later completion, say where annotators do not have sufficient knowledge readily available to fully disambiguate.

For each sentence, not only the resulting preference(s) (or, in rare cases, the conclusion that no correct analysis was available) but also all decisions made by annotators are recorded in the [incr tsdb()] database. Thus, annotator decisions are available as first class data for the semi-automated treebank update procedure introduced in Section 4.4. In a nutshell, semi-automatic updating of the treebank for an enhanced version of the underlying grammar can be achieved by re-applying the recorded disambiguating decisions to a new version of the corpus obtained from re-running the parser on the original data set.

4.2. Representations 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 grammar rules (constructions) used to construct the analysis, (ii) as a traditional phrase structure tree labeled with an inventory of some 50 atomic labels (of the type 'S', 'NP', 'VP' et al.), and (iii) as an underspecified MRS meaning representation. While (ii) will in many cases be similar to the representation found in the Penn Treebank, (iii) subsumes the functor–argument (or tectogrammatical) structure as is advocated in the Prague Dependency Treebank or the German TiGer corpus. Most importantly, however, representation (i) provides all the information required to reconstruct the full HPSG analysis (e.g. using the corresponding version of the HPSG 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 the full analysis and adapting the existing mappings – e.g. the node labeling facilities of the LKB – to their needs.[4] Figures 2 through 4 depict the internal Redwoods encoding and two export representations – labeled constituent trees providing traditional phrase structure and elementary dependency graphs corresponding to functional structure, respectively – derived from existing conversion routines. Labeled phrase structure trees result from reconstructing a derivation (using the relevant grammar) and matching a user-defined set of underspecified feature structure 'templates' against the HPSG feature structure at each node in the tree. In the same spirit, one could apply a set of tree rewrite rules on select parts of the derivation tree (before or after the labeling) in order to map the tree topology into a specific target format, for example, collapsing recursive applications of the head–complement construction on verbal heads in order to convert a binary-branching into a flat verb phrase.

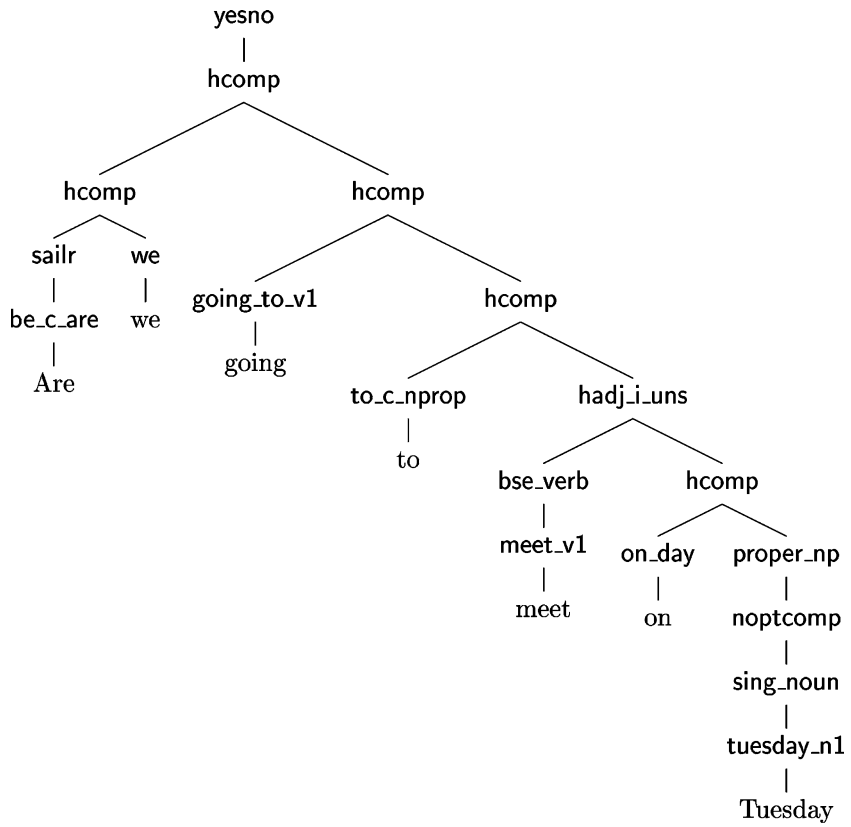

Figure 2. Native Redwoods representation for the sentence *Are we going to meet on Tuesday?* (taken from the current development corpus): the derivation tree is labeled with unique identifiers for grammar rules and lexical entries used to form this analysis.

The elementary dependency graph, on the other hand, is an abstraction from the full MRS meaning representation associated with each full analysis; informally, elementary dependencies correspond to the type of tectogrammatical representations found in the Prague Dependency Treebank and the German TiGer or Dutch Alpino corpora and, likewise, resemble the basic 'grammatical' relations suggested for parser evaluation by Caroll et al. (1998). Given a rich body of MRS manipulation and conversion software, it is relatively straightforward to adapt the type and form of elementary dependencies to user needs or include further information from the full semantic structure (scope constraints, for example).

For evaluation purposes, the existing [incr tsdb()] facilities for comparing across competence and performance profiles can be deployed to gauge results of a (stochastic) parse disambiguation system, essentially using the preferences recorded in the treebank as a 'gold standard' target

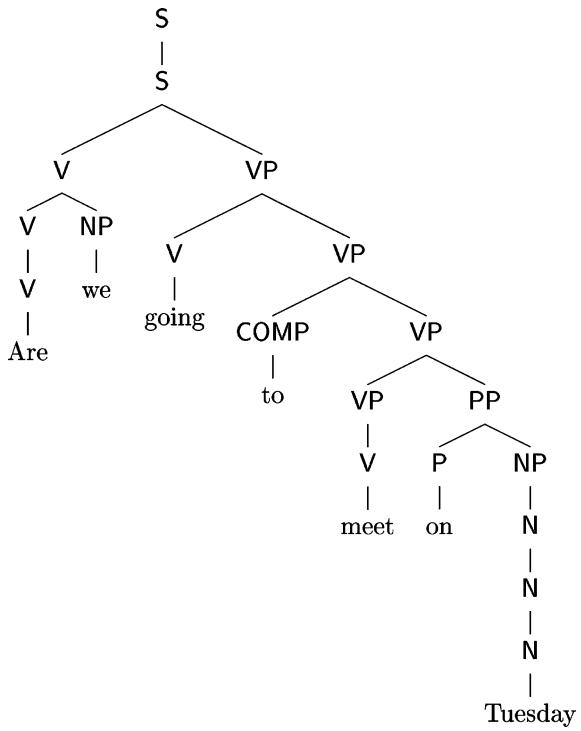

Figure 3. Derived Redwoods representation: phrase structure trees are labeled with user-defined, parameterizable category abbreviations (currently some 57 feature structure templates in the October 2002 version of the LinGO ERG).

```
_3:{
    _3:int_rel[SOA e2:_meet_v_rel]
    e2:_meet_v_rel[ARG1 x4:pron_rel]
    _1:def_rel[BV x4:pron_rel]
    e14:_on_temp_rel[ARG e2:_meet_v_rel, ARG3 x12:dofw_rel]
    _2:def_np_rel[BV x12:dofw_rel]
    x12:dofw_rel[NAMED :tuesday]
  }
```

Figure 4. Another derived Redwoods encoding: an elementary dependency graph extracted from MRS meaning representation associated with the underlying derivation tree. The nodes are comprised of MRS relations, of which most are contributed by lexical entries but also allowing for semantic contributions from non-lexical elements in the full HPSG derivation (e.g. the representation of illocutionary force by virtue of MRS messages). Arcs of the dependency graph are labeled by uninterpreted MRS role labels (ARG1, SOA et al.) which could be assigned user-level interpretations as, for example, thematic roles relative to the lexicon and various MRS relation types.

for comparison. While the concept of a meta-treebank of the type proposed here has been explored in earlier research (e.g. the AMALGAM project at Leeds University in the UK; Atwell, 1996), previous approaches to the dynamic mapping of treebank representations have built on a static, finite set of hand-constructed mappings.

4.3. Scope and current state of play

Starting in June 2001, the project decided to initially explore domains for which (i) sufficient amounts of real-world data were readily available, (ii) the LinGO ERG could be expected to exhibit broad and accurate coverage, and (iii) a lot of labor had been expended earlier in constructing (and maintaining) hand-built parse selection heuristics. Accordingly, the first 12,000 trees to be hand-annotated in Redwoods format are taken from transcribed face-to-face dialogues in an appointment scheduling and travel arrangement domain, viz. a representative sample of the data produced in the VerbMobil Wahlster, 2000 machine translation project. Corpora of some 50,000 such utterances are publicly available and have already been studied among researchers world-wide in the field.

Table I summarizes the current Redwoods annotation status, reflecting three consecutive development phases. Of a total of some 12,000 hand-segmented turns in dialogues recorded on four CDs, some 3600 are flagged as fragment (incomplete) utterances and close to 700 as simply ungrammatical. Corpora containing strongly ungrammatical input are generally problematic for the Redwoods approach, as it makes the basic assumptions that (i) the underlying linguistic grammar is precise and maintains a sharp (and therefore idealized) distinction between well-formed vs. ill-formed utterances and (ii) the treebank is entirely constructed from analyses provided by the grammar, such that the grammar and associated tools can be used to extract syntacto-semantic information. Fragmentary, non-sentence utterances also present a practical problem, though not in principle: although the LinGO ERG includes facilities to accept strings like *Wednesday, in the afternoon?* or *Monday and Tuesday of next week.* (essentially, through relaxation of the grammar start symbol), enabling this fragment mode also generates a large number of spurious analyses for full sentences – accepting each imperative as a possible verb phrase fragment, for example. For the early Redwoods development phase it was therefore decided to exclude both fragments and ungrammatical utterances from the annotation (but not the corpus, of course). Although one immediate consequence of this decision is that complete linguistic information is only available for a subset of the corpus (which may limit its utility for certain users), using the corpus as training material for stochastic disambiguation models

is not hindered, as there is, of course, no parse selection problem for utterances with no analysis in the LinGO ERG.[5]

As shown in Table I, the Redwoods treebank has undergone three revision stages to date, dubbed 1st, 2nd, and 3rd Growth, respectively. Where the first two versions both use the June 2001 version of the LinGO ERG, the most recent 3rd Growth version reflects a much later version of the grammar, viz. the ERG as of October 2002 (see Section 4.4 below). Of the 8049

Table I. Redwoods development status as of January 2003: four sets of transcribed and hand-segmented dialogues have been annotated

	all parses			active = 0			active = 1			active > 1		
	♯	‖	×	♯	‖	×	♯	‖	×	♯	‖	×
1st Growth												
VM_6	2422	7·7	32·9	218	8·0	9·7	1910	7·0	7·5	80	10·0	23·8
VM_{13}	1984	8·5	37·9	175	8·5	9·9	1491	7·2	7·5	85	9·9	22·1
VM_{31}	1726	6·2	22·4	164	7·9	8·0	1360	6·6	5·9	61	10·1	14·5
VM_{32}	608	7·4	25·6	51	10·7	54·4	549	7·9	19·0	7	10·4	20·6
Total	6740	7·5	31·0	608	8·3	13·0	5310	7·1	8·3	233	10·0	20·7
2nd Growth												
VM_6	2422	7·7	32·9	234	8·6	24·7	2088	7·6	31·2	100	11·0	89·2
VM_{13}	1984	8·5	37·8	204	9·5	43·7	1670	7·9	24·6	110	10·9	76·0
VM_{31}	1726	6·1	22·4	190	9·0	41·6	1465	7·0	19·3	71	10·2	35·2
VM_{32}	608	7·4	25·6	51	10·7	54·4	552	7·9	23·0	5	12·2	27·2
Total	6740	7·5	31·0	679	9·1	37·4	5775	7·6	25·5	286	10·8	69·6
3rd Growth												
VM_6	2706	7·7	46·7	216	9·4	63·5	2484	8·3	43·5	6	15·8	757·8
VM_{13}	2279	8·5	61·9	248	10·8	80·5	2028	8·7	59·5	3	15·5	198·0
VM_{31}	1967	6·2	27·9	216	10·1	95·9	1746	7·5	30·8	5	8·4	20·8
VM_{32}	697	7·5	53·2	16	11·8	57·7	681	8·4	53·2	0	0·0	0·0
Total	7649	7·5	47·0	696	10·2	79·5	6939	8·2	45·9	14	12·9	388·2

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 ('‖'), and structural ambiguity ('×'), 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), and (iii) where full disambiguation was not accomplished through the first round of annotation (more than one active tree). Around six per cent of massively ambiguous sentences had not been annotated in the 1st Growth release and, of the four data sets, only VM_{32} had been double-checked by an expert grammarian and (almost) completely disambiguated; therefore it exhibits a somewhat higher degree of phrasal ambiguity in the 'active = 1' subset.

grammatical sentences in the four data sets, the grammar has coverage (i.e. derives at least one analysis) for around 84 and 95% (in the June 2001 and October 2002 versions, respectively) of the corpus. The relatively short string length – less than eight words per sentence, on average – reflects the spoken dialogue nature of the corpus, while the average ambiguity rate of 31 analyses per sentence (47 in the 3rd Growth) in part is a function of sentence length, of course, but also confirms the expectation that a hand-built precision grammar like the LinGO ERG exhibits a radically different ambiguity rate than, say, grammars derived from the Penn Treebank.

The main differences between the first two Redwoods releases are best seen in the column labeled 'active = 1' in Table I, i.e. the part of the corpus for which annotation resulted in full disambiguation (which, presumably, is also the subset of the available data that is immediately most relevant to experimentation with stochastic parse selection models; see Toutanova and Manning, 2002). While in the 1st Growth release a residue of some 6% of massively ambiguous items was left unannotated, the 2nd Growth version included annotations for all sentences, resulting in larger average ambiguity per sentence across the board; at the same time, a relatively large number of items still had to be left partially disambiguated (the 'active > 1' column) in the 2nd Growth, mostly because of a deficiency in the June 2001 version of the LinGO ERG that interacts badly with the Redwoods discriminants philosophy.[6] Conversely, moving to a greatly enhanced grammar in October 2002, the 3rd Growth recovers most of the items left partially resolved earlier – which tend to be highly ambiguous – and also adds grammatical coverage; part of the increased overall coverage, however, is achieved by allowing the parser to explore a larger search space (such that fewer items are rejected due to resource limitations) which, in turn, tends to add more highly ambiguous sentences into the active part of the corpus. When comparing across identical subsets of the 2nd and 3rd Growth versions (as is shown in Table II of Section 4.4 below), in fact, it is revealed that ambiguity has not increased nearly as drastically as might be suggested by the overall average. In a sense, a fairly small number of massively ambiguous outliers in the distribution inflates the (arithmetic) averages in Table I; therefore, we are optimistic that the parse selection problem has not been made drastically more difficult in more recent Redwoods versions.

While annotation of further data, specifically in fragment utterances, and inter-annotator cross-validation continue, the current development snapshot (both 1st and 3rd Growth) of the treebank is publicly available already. Work on stochastic parse selection models for the Redwoods treebank is underway, so far obtaining an *exact match* parse selection accuracy of above 80% from a combination of methods applied to the Redwoods derivation trees and elementary dependency graphs (see Figures 2 and 4, respectively); details on Redwoods parse selection results are reported by

Table II. Quantitative assessment of grammar evolution between June 2001 (underlying the 1st and 2nd Growth versions of Redwoods) and October 2002 (3rd Growth)

	Jun-01	Oct-02	Δ
Appropriate features	148	149	−6% +7%
Type hierarchy (excluding lexicon)	3062	3895	+27%
Grammar rules (including lexical rules)	86	94	−11% +26%
Lexical types ('parts of speech')	400	580	+45%
Semantic relations ('predicates')	5406	6162	+14%
Lexical entries	8135	9954	+22%
Lines of source (excluding lexicon)	25847	32199	+25%

The column labeled Δ 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.

Toutanova and Manning (2002) and Open et al. (2002). For a follow-up phase of the Redwoods initiative, we have moved into a different domain and text genre – annotating an 8000-item e-commerce email corpus – and also consider more formal, edited text taken from newspaper text or another widely available on-line source.

4.4. Treebank maintenance and evolution

Among the most challenging research aspects of the Redwoods initiative was the search for a methodology for automated updates of the treebank, in order to keep track with the continuous evolution of the underlying linguistic framework and of the LinGO English Resource Grammar. We believe that we have found an innovative procedure that – again crucially building on the notion of elementary linguistic discriminators – allows us to maintain the treebank in synchronization with ongoing grammar development work, with minimal manual effort. In fact, our semi-automatic update procedure helps grammarians to identify and isolate effects of changes made in the grammar and, thus, could be integrated into the regular grammar engineering and regression test routines.

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 there is reason to expect that at least some of the disambiguating decisions can be re-used. 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. These entailed decisions are recorded at annotation time as well and – in conjunction with the small amount of redundancy already present in the use of partly overlapping discriminants already (see the discussion in Section 4.1 above) – make the record keeping of 'disambiguating potential' highly redundant.

A complete, 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 intersect it with recorded decisions for this sentence in the earlier corpus;
(3) *manual resolution* a user-supplied predicate decides, for each item, whether the automated update was successful and complete; the remaining items require further annotator inspection and manual disambiguation.

After close to a decade of work on the LinGO ERG, it can be assumed that the basic phrase structure inventory and granularity of lexical distinctions have stabilized to a certain degree. However, 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 seems to be of no concern in practice: the grammar 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 the 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 writing 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 update cycle that resulted in the Redwoods 3rd Growth version. For this exercise to be a strong measure of how much of the 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 LinGO ERG was very actively used in building a commercial product (for automated email response) and adapted from the original VerbMobil domain to e-commerce and financial transaction emails. Accordingly, the 'distance' between the two versions of the grammar used in the treebank update reported here is exceptionally large. Table II attempts to compile a summary of changes made to the grammar between June 2001 and October 2002; although it is in general hard to quantify grammar evolution and compare across grammar versions, some of the measures reported in Table II immediately pertain to the type of information used in Redwoods discriminants: the inventory of grammar rules, lexical types, and semantic relations determines the range of elementary properties used in the Redwoods approach. Between the two versions of the LinGO ERG in question, we observe differentials of 37, 45, and 14%, respectively, for these three central measures. Clearly, the scope of the update problem is much bigger in this scenario than would usually be expected, if one were to keep the treebank in line with the grammar at least a few times each year, say (as illustrated below in a second, smaller experiment).

Practical update results are summarized in Table III, showing a number of relevant measures. The update procedure itself provided important 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 dialogues, while the remaining two were only updated once the grammarian had converged on the final version of the LinGO 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 III (labeled 'VM_{13+31}') and shows that close to 60% 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 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

Table III. Quantitative summary of semi-automated update process on ambiguous items: the table reflects the amount of manual intervention for two distinct update scenarios, viz. one update after 18 months of grammar evolution and a second after three weeks (labeled 'VM_{13+31}' and 'VM_{6+32}', respectively)

		original		matches		update			final	
Aggregate	Items	in	out	yes	no	in	out	new	in	out
	♯	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
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

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.

decisions could be re-used (for an overall average re-use ratio of 46%), 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 the stable average ambiguity rate across the two data sets (see also the discussion of Table I above) even though – given the fairly drastic revisions in the grammar – no two derivations would yield an exact match. The lower part of Table III (labeled 'VM_{6+32}'), 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 2288 ambiguous items required a total of 130 additional annotator decisions. Given the full integration of the update procedure and annotation environment, we conjecture that a full treebank update (across reasonably similar grammar

versions) can be completed in a matter of minutes or hours and should become part of the standard regression testing and release cycle for the LinGO ERG.

5. Related Work

To the best of our knowledge, no prior research has been conducted exploring both the linguistic depth, flexibility in available information, and dynamic nature of treebanks as proposed presently. 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 the light of the successful LinGO open-source repository, it seems vital that both the treebank and associated processing schemes and stochastic models be made widely available.

An on-going initiative at Rijksuniversiteit Groningen (NL) is developing a bank of dependency structures (Mullen et al., 2001; van der Beek et al., 2002), as they are derived from an HPSG-like grammar of Dutch (Bouma et al., 2001). While the general approach resembles the Redwoods initiative (specifically the discriminator-based method used in selecting trees from the set of analyses proposed by the grammar; the LKB tree selection tool was originally developed by Malouf, after all), there are three important differences. First, the Groningen decision to compose the treebank from dependency structures commits the resulting resource to a single stratum of representation, tectogrammatical structure essentially, and thus eliminates some of the flexibility in extracting various types of linguistic structure that the Redwoods architecture affords. Second, and in a similar vein, recording dependency structures means that the (stochastic) disambiguation component has to consider two syntactically different analyses equivalent whenever they project identical dependency structures; hence, there is a mismatch of granularity between the disambiguated treebank structures and the primary structures (i.e. derivation trees) constructed by the grammar. Finally, the Groningen initiative is making the assumption that the dependency structures, once they are stored in the treebank, are correct and do not change over time (or as an effect of grammar evolution); disambiguating decisions made by annotators are *not* recorded in the treebank, nor does the project expect to dynamically update the treebank with future revisions of the underlying grammar.

Another closely related approach is the work reported by Dipper, 2000, essentially the application of a broad-coverage LFG grammar for German to constructing tectogrammatical structures for the TiGer corpus. While many

of the basic assumptions about the value of a systematic, broad-coverage grammar for treebank construction are shared, the strategy followed by Dipper (2000) exhibits the same limitations as the Groningen initiative: the TiGer target representation, still, is mono-stratal and the approach to hand-disambiguation and subsequent transfer of result structures into the TiGer corpus loses the linkage to the original analyses and basic properties used in the disambiguation, and hence the potential for dynamic adaptation of the data or automatic updates.

Finally, for the BulTreeBank initiative at Sofia (Bulgaria) and Tübingen (Germany) which, in turn, appears to share a number of goals with our work, it is too early yet to draw a technical comparison (Simov et al., 2002).

Acknowledgements

The Redwoods initiative is part of the LinGO Laboratory at CSLI and many people, both at Stanford and at partner sites, have contributed to its design and results so far. Ezra Callahan performed the first round of treebank annotation and, with great patience and constructive criticism, helped improve the ergonomics of the annotation process. Ivan Sag, Tom Wasow, Emily Bender, Tim Baldwin, John Beavers, and Kathryn Campbell-Kibler have all participated in our regular 'tree conferences', helping annotators select parses and offering productive critiques on analyses provided by the LinGO grammar. Ann Copestake, John Carroll, Rob Malouf, and Stephan Oepen are the main developers of the LKB, tree comparison tool, and [incr tsdb()] software packages from which the Redwoods treebanking environment has been built and, in various capacities, have influenced the Redwoods approach significantly; besides supplying part of the initial design and implementation, Rob Malouf has been especially supportive in relating our own experience to the closely related work at Rijksuniversiteit Groningen. During a three-month visit to Stanford, Stuart Shieber was among the driving forces for applications of the existing development version of the treebank, helping us build and fine-tune suitable stochastic parse selection models. Redwoods has been partially funded by an internal opportunity grant from CSLI Stanford and by a donation from YY Technologies (Mountain View, CA). More recent work is in part supported through participation in the EU-funded Deep-Thought consortium and the ROSIE project funded by Scottish Enterprise within the Edinburgh–Stanford Link. Our Edinburgh colleagues – Jason Baldridge, Alex Lascarides, Miles Osborne, and David Schlangen – have since helped improve and extend the Redwoods resources significantly.

Notes

[1] All LinGO resources – the software, grammars, and preliminary Redwoods versions – are available for public download from '`http://lingo.stanford.edu/`'.

[2] The general Redwoods methodology is, of course, language-independent, and LinGO collaborators in Germany, Japan, and Norway have already started the construction of Redwoods-style treebanks for their languages (termed *Eiche*, *Hinoki*, and *Bjørk*, respectively).

[3] It might even turn out that a dynamic inventory of discriminants with increasing complexity will benefit the annotation process. In particular, for highly ambiguous items, it may be feasible to reduce the parse forest in an initial annotation phase by means of unlabeled 'bracketing' discriminants only (which, in turn, could be seeded from a reliable phrase boundary detector if such a tool was available) and only in a later annotation phase increase discriminant granularity to the degree required for full disambiguation. Another scenario we are exploring involves a successive reduction of the packed parse forest itself, i.e. the unfolding and disambiguation of packing nodes, as they correspond to local ambiguity.

[4] Pre-existing, external software packages like the freely available *Tree Processor* (see '`http://www.cis.upenn.edu/~dchiang/treep.html`') might supply other candidate mapping tools, although unlike the LKB this approach would not have direct access to the type system underlying the.

[5] Work on filling in Redwoods annotations for the fragment utterances is underway now as a separate annotation cycle. Given the availability of grammaticality and fragment annotations in the base corpus, a similar approach could be pursued to include analyses for ungrammatical utterances: the LinGO ERG, when used in connection with the PET parser Callmeier, 2000, also provides devices to process ill-formed input and, using specialized 'robustness' rules, also derive analyses for such utterances.

[6] The problem in the grammar was that, for nominal heads with multiple optional complements, the grammar would admit spurious ambiguity as to whether all optional complements were explicitly discharged through recursive applications of the same non-branching rule or not. Given the inventory of discriminants sketched in Section 4.1 above, (spurious) ambiguity of this type cannot be resolved straightforwardly.

References

Agresti, A. (1990) *Categorical Data Analysis.* John Wiley & Sons

Atwell, E. (1996) Comparative Evaluation of Grammatical Annotation Models. In Sutcliffe, R., Koch, H.-D., McElligott, A. (eds.), *Proceedings of the Workshop on Industrial Parsing of Software Manuals*, Rodopi, Amsterdam, The Netherlands, pp. 25–46.

Bouma, G., van Noord, G., Malouf, R. (2001) Alpino. Wide-Coverage Computational Analysis of Dutch. In Daelemans, W., Sima-an, K., Veenstra, J., Zavrel, J. (eds.), *Computational Linguistics in the Netherlands.* Rodopi, Amsterdam, The Netherlands, pp. 45–59.

Callmeier, U. (2000) PET — A platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering,* 6/1 (Special Issue on Efficient Processing with HPSG), pp. 99–108.

Carroll, J., Briscoe, E., Sanfilippo, A. (1998) Parser evaluation: a survey and a new proposal. In *Proceedings of the 1st International Conference on Language Resources and Evaluation,* Granada, Spain, pp. 447–454.

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*, Providence, RI, pp. 598–603.

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*, Madrid, Spain, pp. 16–23.

Copestake, A. (1992) The ACQUILEX LKB. Representation Issues in Semi-Automatic Acquisition of Large Lexicons. In *Proceedings of the 3rd ACL Conference on Applied Natural Language Processing* Trento, Italy, pp. 88–96.

Copestake, A. (2002) *Implementing Typed Feature Structure Grammars.* CSLI Publications, Stanford, CA.

Copestake, A., Flickinger, D., Sag, I. A., Pollard, C. (1999) *Minimal Recursion Semantics. An Introduction.* In preparation, CSLI, Stanford, CA.

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,* Luxembourg, pp. 56–64.

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

Hajic, J. (1998) Building a syntactically annotated corpus. The Prague dependency treebank In *Issues of Valency and Meaning*, Karolinum, Prague, Czech Republic, pp. 106–132.

Harris T. E. (1963) *The Theory of Branching Processes,* Springer, Berlin, Germany.

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,* College Park, MD, pp. 535–541.

Kiefer, B., Krieger, H.-U., Carroll, J., Malouf, R. (1999) A Bag of Useful Techniques for Efficient and Robust Parsing. In *Proceedings of the 37th Meeting of the Association for Computational Linguistics,* College Park, MD, pp. 473–480.

King, T. H., Dipper, S., Frank, A., Kuhn, J., Maxwell, J. (2000) Ambiguity management in grammar writing In *Workshop on Linguistic Theory and Grammar Implementation.* Birmingham, UK, pp. 5–19.

Malouf, R., Carroll, J., Copestake, A. (2002) Efficient feature structure operations without compilation. In Oepen, S., Flickinger, D., Tsujii, J., Uszkoreit, H., (eds.), *Collaborative Language Engineering. A Case Study in Efficient Grammar-based Processing,* CSLI Publications, Stanford, CA.

Marcus, M. P., Santorini, B., Marcinkiewicz, M. A. (1993) Building a large annotated corpus of English. The Penn Treebank. *Computational Linguistics 19*, pp. 313–330.

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

Müller, S., Kasper, W. (2000) HPSG Analysis of German. In Wahlster, W., (ed.), *Verbmobil. Foundations of Speech-to-Speech Translation* (Artificial Intelligence ed.), Springer, Berlin, Germany, pp. 238–253.

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,* Trento, Italy, pp. 183–194.

Oepen, S., Carroll, J. (2000) Performance Profiling for Parser Engineering. *Natural Language Engineering 6/1* (Special Issue on Efficient Processing with HPSG), pp. 81–97.

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.

Pollard, C., Sag, I. A. (1994) *Head-Driven Phrase Structure Grammar.* The University of Chicago Press and CSLI Publications, Chicago, IL and Stanford, CA.

Simov, K., Osenova, P., Slavcheva, M., Kolkovska, S., Balabanova, E., Doikoff, D., Ivanova, K., Simov, A., Kouylekov, M. (2002) Building a Linguistically Interpreted Corpus of Bulgarian. The BulTreeBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation,* Canary Islands, Spain, pp. 1729–1736.

Skut, W., Krenn, B., Brants, T., Uszkoreit, H. (1997) An Annotation Scheme for Free Word Order Languages. In *Proceedings of the 5th ACL Conference on Applied Natural Language Processing,* Washington, DC.

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.

van der Beek, L., Bouma, G., Malouf, R., van Noord, G. (2002) The Alpino Dependency Treebank. In Theune, M., Nijholt, A., Hondorp, H. (eds.), *Computational Linguistics in the Netherlands.* Rodopi, Amsterdam, The Netherlands.

Wahlster, W. (ed.). (2000) *Verbmobil. Foundations of Speech-To-Speech Translation.* Springer, Berlin, Germany.