Sustainable Development and Refinement of Complex Linguistic Annotations at Scale

Dan Flickinger, Stephan Oepen, and Emily M. Bender

1 Introduction

Linguistic annotation projects in general serve two functions: On the one hand, a great deal can be learned about language structure and language use by applying an operationalized set of categories to running speech or text. On the other hand, the resulting resources can be valuable for both engineering goals (training machine learners) and scientific ones (supporting data exploration). Because languages involve subsystems which are both intricate and interconnected, annotations which are rich enough to represent complete analyses of utterances at multiple levels of linguistic structure are more valuable, both in the process of their creation and in the resulting resource. However, the more complex the linguistic annotations, the more difficult it is to produce them consistently at interesting scales.

In this paper, we argue that developing complex linguistic annotations calls for an approach which allows for the incremental improvement of existing annotations by encoding all manual effort in such a way that its value is preserved and enhanced even as the resource is improved over time. The manual effort includes both annotation design and disambiguation. In the case of syntactico-semantic annotations, the former can be encoded in a machine-readable grammar and the latter as a series of decisions made at a level of granularity which supports both efficient human disambiguation and later machine re-use of the individual decisions. These two ways of storing the manual effort involved in annotations are central to the Redwoods (Oepen, Flickinger, Toutanova, & Manning, 2004) approach to treebank construction, described in § 2 and § 3 below. We believe that the general approach can be

Dan Flickinger
Stanford University, e-mail: danf@stanford.edu

Stephan Oepen
University of Oslo, e-mail: oe@ifi.uio.no

Emily M. Bender
University of Washington, e-mail: ebender@uw.edu

applied beyond syntactico-semantic annotation to any annotation project where the design of the representations can be encoded as a grammar, and thus we frame our methodological discussion in § 4 in terms of incremental improvement, with syntactico-semantic annotations as a case study. Other projects beyond Redwoods have taken a similar approach, and these are reviewed in § 5.

There is of course still a long way to go if the ultimate goal is complete, comprehensive annotations at all levels of linguistic structure over a truly representative sample of texts for even a single language (English, in the case of Redwoods). Some of the challenges ahead are addressed in § 6. As we think about the progress of the field so far and look ahead to upcoming challenges, we propose a thought experiment: Imagine the ideal annotated resource, comprising if not a comprehensive collection of linguistic data then at least a very large sample representing the gamut of genres and registers, including academic writing, literature, and news articles, but also social media content, caretaker speech, song lyrics, pillow talk, and all the other myriad ways in which speakers use our language. This collection of text (and transcribed speech) would then have full annotation, including morphology, syntax, compositional semantics, pragmatics, prosody, word sense, and more. All of those annotations would be consistent across the entire (very very large) corpus, free of errors, fully documented, and freely available. We will argue in this paper that the sort of incremental improvement of annotated resources enabled by the Redwoods approach—the selection by human annotators among representations produced by machine using a grammar created in turn in a rule-based fashion—is critical to moving along the path towards that ideal.

2 Background: Redwoods Motivation & History

At the core of our methodological reflections in this chapter are two linguistic resources that have been under continuous development for more than a decade now. First, the LinGO English Resource Grammar (ERG; Flickinger, 2000) is an implementation of the grammatical theory of Head-Driven Phrase Structure Grammar (HPSG; Pollard & Sag, 1987, 1994) for English, i.e. a computational grammar that can be used for parsing and generation. Development of the ERG started in 1993, building conceptually (if not practically) on earlier work on unification-based grammar engineering for English at Hewlett Packard Laboratories (Gawron et al., 1982). The ERG has continuously evolved through a series of R&D projects (and two commercial applications) and today allows the grammatical analysis of running text across domains and genres. The hand-built ERG lexicon of some 38,000 lemmata aims for complete coverage of function words and open-class words with 'non-standard' syntactic properties (e.g. argument structure). Built-in support for light-weight named entity recognition and an unknown word mechanism combining statistical PoS tagging and on-the-fly lexical instantiation for 'standard' open-class words (e.g. names or non-relational common nouns and adjectives) typically enable the grammar to derive complete syntactico-semantic analyses for 85–95 percent of

all utterances in standard corpora, including newspaper text, the English Wikipedia, or bio-medical research literature (Flickinger, Zhang, & Kordoni, 2012; Flickinger, Oepen, & Ytrestøl, 2010; Adolphs et al., 2008). Parsing times for these data sets average around two seconds per sentence, i.e. time comparable to human production or comprehension.

Second, since around 2001 the ERG has been accompanied by a selection of development corpora, where for each sentence an annotator has selected the intended analysis among the alternatives provided by the grammar, or has recorded that no appropriate analysis was available (in a given version of the grammar). This derived resource is called the LinGO Redwoods Treebank (Oepen et al., 2004). For each release of the ERG, a corresponding version of the treebank has been produced, manually validating and updating existing analyses to reflect changes in the underlying grammar, as well as 'picking up' analyses for previously out-of-scope inputs and new development corpora. In mid-2013, the current version of Redwoods (dubbed Ninth Growth) encompasses gold-standard ERG analyses for some 85,400 utterances (or close to 1.5 million tokens) of running text from half a dozen different genres and domains, including the first 22 sections of the venerable Wall Street Journal (WSJ) text in the Penn Treebank (PTB; Marcus, Santorini, & Marcinkiewicz, 1993).

The original motivation to start treebanking ERG analyses was to enable training discriminative parse selection models, i.e. a conditional probability distribution to rank ERG analyses, and to thus approximate the abstract notion of the 'intended' analysis of an utterance as the statistically most probable one (Abney, 1997; Johnson, Geman, Canon, Chi, & Riezler, 1999). For this purpose, the treebank should disambiguate at the same level of linguistic granularity as is maintained in the grammar, i.e. encode the same (or closely comparable) grammatical distinctions; external resources such as the PTB are not sufficient for this purpose, since they do not make the same range of distinctions as the ERG. Furthermore, to train discriminative (i.e. conditional) statistical models, both the intended as well as the dispreferred analyses are needed. For these reasons, treebanking ERG analyses was a practical necessity to facilitate probabilistic disambiguation.

In Redwoods, the treebank is built exclusively from ERG analyses, i.e. full HPSG syntactico-semantic signs. Annotation in Redwoods amounts to disambiguation among the candidate analyses proposed by the grammar (identifying the intended parse) and, of course, analytical inspection of the final result. To make this task practical, a specialized tree selection tool extracts a set of what are called discriminants from the complete set of analyses. Discriminants encode contrasts among alternate analyses—for example whether to treat a word like *record* as nominal or verbal, or where to attach a prepositional phrase modifier. Whereas picking one full complete analysis (among a set of hundreds or thousands of trees) would be daunting (to say the least), the isolated contrasts presented as discriminants are comparatively easy to judge for a human annotator, even one with only a limited understanding of grammar internals.

Discriminant-based tree selection was first proposed by Carter (1997) and has since been successfully applied to a range of grammatical frameworks and grammar

engineering initiatives (see § 5 below). But to the best of our knowledge Redwoods remains the longest-running and most comprehensive such effort, complementing the original proposal by Carter (1997) with the notion of *dynamic* treebanking, in two senses of this term. First, different views can be projected from the multi-stratal HPSG analyses at the core of the treebank, highlighting subsets of the syntactic or semantic properties of each analysis, e.g. HPSG derivation trees, more conventional phrase structure trees, full-blown logical-form meaning representations, variable-free elementary semantic dependencies, or even reductions into just bi-lexical syntactic or semantic dependencies (see § 3 below). Second, a dynamic treebank can be extended and refined over time. Dynamic extension of a treebank refers to the ease with which it can be expanded to include data from additional texts (including new genres) while maintaining consistency of annotations. Dynamic refinement refers to the ability to add detail to the linguistic analyses (through refinement of the underlying grammar) and do systematic error correction while minimizing any loss of manual input from previous annotation cycles.

The Redwoods Treebank achieves dynamic extension by locating the bulk of the linguistic analytical (manual) effort in the development of the English Resource Grammar. Although we can by no means quantify precisely the effort devoted to ERG and Redwoods development to date, we estimate that around 25 person years have been accumulated between 1993 and 2013. In contrast with encoding linguistic analyses in annotation guidelines, encoding them in a grammar simplifies their application to new text to a task that can be carried out by a machine, and thus applied to new texts inexpensively.[1]

We achieve dynamic refinement by pairing the resource grammar–based approach to encoding linguistic knowledge with a cumulative approach to discriminant-based treebanking for selecting linguistic analyses in context: the treebank records not only the analysis ultimately selected (and validated) by the annotator, but also all annotator decisions on individual discriminants, which 'signpost' the disambiguation path leading to the preferred analysis. This makes updating the treebank to a newer release of the ERG comparatively cost-effective: the vast majority of annotator decisions can be reused, i.e. re-applied automatically to the set of analyses licensed by the revised grammar. In addition, because there is considerable redundancy in the recorded information, it will often be the case that 'fresh' annotator decisions on discriminants are only required where grammar evolution has genuinely enlarged the space of candidate analyses, including of course making available a good analysis for previously untreated inputs. Thus when the grammar is updated to handle new phenomena or refine e.g. the semantic representation associated with a previously analysed phenomenon, the production of a new treebank version incorporating these refinements is eminently practical, and has been demonstrated repeatedly over the past decade for the ERG and the Redwoods Treebank.

[1] A grammar is never complete, however, and new texts always hold the promise of new linguistic phenomena to investigate. The ability to process the text with a grammar encoding the existing analyses makes it much easier to discover those which are not yet covered by the grammar, even as they become ever less frequent.

Furthermore, we find that treebanking, rather than being a distraction to grammar development, in fact supports it: as Oepen et al. (2004) argue, this update procedure contributes useful information to the grammar development cycle. We bring this mutual feedback loop between grammar engineering and annotation into focus in §4 below, describing the ongoing cycle of the refinement of the formally encoded repository of general grammatical knowledge, on the one hand, and the in-depth study of individual linguistic examples and their candidate analyses, on the other.

Interestingly, there is a very clear tendency for the treebank-related tasks to take a steadily growing proportion of total development effort. When preparing the most recent release of the ERG (dubbed 1212) and associated treebank, we estimate that around two thirds of the time invested over the course of a year went into updating analyses for existing treebanked corpora, with the other third spent on the grammar itself, augmenting linguistic coverage, reducing spurious ambiguity, making semantic analyses more consistent, and pursuing greater efficiency in processing. The concurrent addition of the WSJ annotations alongside the 1212 release of the ERG will inevitably increase the treebank maintenance costs for the next release of the grammar. Nonetheless, the effort of treebanking remains a valuable part of the grammar development process, even as it takes a larger and larger proportion of development time, as the larger the treebank, the more sensitive it is as a regression testing tool. We return to these issues in §4.

3 Redwoods: Annotation Contents

To give a sense of the degree of complexity of annotation that this approach can support, this section provides an extended discussion of a relatively short yet interestingly complex example sentence, given in (1).[2]

(1) An analogous technique is almost impossible to apply to other crops.

The 1212 version of the ERG finds 15 complete analyses of this string. Among that forest of analyses, a typical discriminant-based disambiguation path would lead to one analysis with three annotation decisions, for example solely through lexical disambiguation: picking the semantically vacuous particle *to*, *impossible* as a *tough*-adjective, and the predicative copula (rather than the identity copula, with an extracted NP complement in this case). In addition to recording these discriminant choices, the treebank stores the ERG derivation tree associated with the selected analysis, shown in Figure 1. Here, tree nodes (above the preterminals) are labelled with HPSG constructions, e.g. instances of the subject–head, specifier–head, and head–complement types. The labels of preterminal nodes are fine-grained lexical categories, called ERG lexical types, which complement classical parts of

[2] This example is an adaptation of a sentence that appears in the WSJ portion of the PTB, as well as in the much smaller Cross-Framework Parser Evaluation Shared Task (PEST) corpus discussed by Ivanova, Oepen, Øvrelid, and Flickinger (2012).

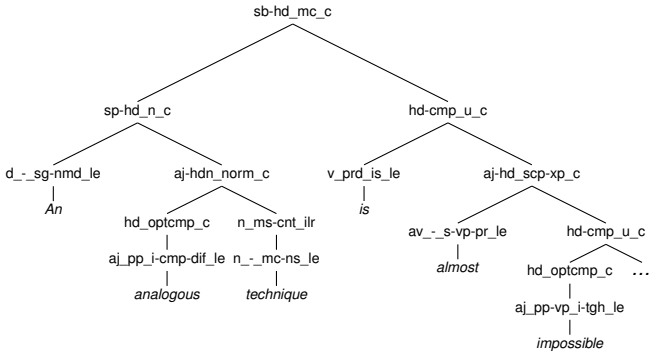

Fig. 1 ERG derivation tree for example (1).

speech with additional grammatical distinctions, for example argument structure or the distinction between count, mass, and proper nouns. This derivation serves as a 'recipe' which can be used in combination with the grammar to regenerate the full HPSG analysis. That analysis is in fact a very large feature-structure, including 3241 feature–value pairs. The feature structure encodes a wide variety of information, some of which is most relevant to grammatical processing (constraints on well-formed structures). Other information more relevant to downstream processing includes syntactic constituent structure; morphosyntactic and morphosemantic features associated with every constituent including part of speech, person, number, gender; syntactic dependency structure; semantic dependency structure; and partial information about scopal relations in the semantics.

Figure 2 shows a partial view of the feature structure associated with the PP node yielding the substring *to other crops* in (1). The feature geometry adopted in the ERG and reflected in Figure 2 largely follows established HPSG conventions for grouping the feature–value pairs into substructures. At the highest level, we see the division into CAT and CONT, which encode syntactic ('category') and semantic ('content') information, respectively. The information under CAT describes a constituent headed by a preposition ([HEAD *prep*]) which has picked up any complements it requires ([VAL|COMPS $\langle\ \rangle$]), is able yet to combine with a specifier (given the non-empty value of SPR), and is prepared to modify a constituent of the type described in its MOD value. That is, this PP is suitable as a modifier of verbal, adjectival, or other prepositional phrases that have in turn already satisfied their own complement requirements. However, in the selected analysis of this example, the PP is picked up as a complement of the verb *apply*, and does not function as a modifier.

The semantic portion of this feature structure, under CONT, describes the contribution that this constituent will make to the semantics of the sentence overall (in the format of Minimal Recursion Semantics (MRS; Copestake, Flickinger, Pollard, & Sag, 2005), and provides the pointers into that contribution required for its composition with the semantic contribution of the rest of the sentence. More specifically, the value of the feature RELS is a list of elementary predications (described through

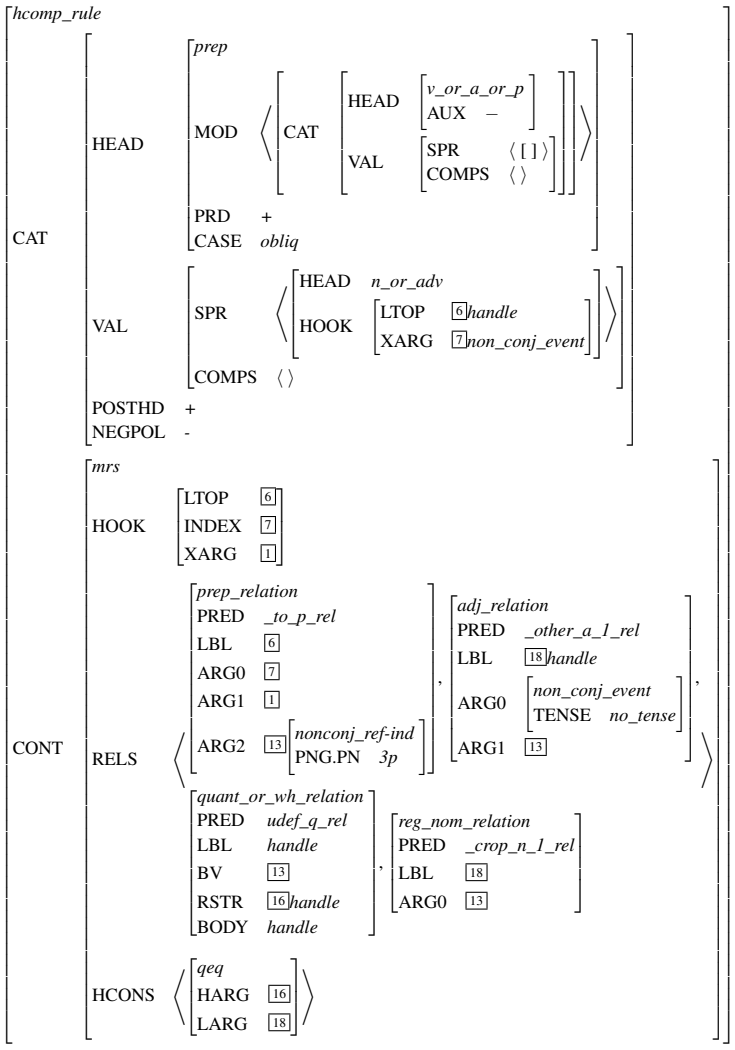

Fig. 2 Partial feature structure for PP *to other crops*

typed feature structures) linked together through shared values, each contributed by a lexical entry or phrase structure rule involved in the construction of the PP or its sub-constituents. The value of the feature HOOK provides pointers to values of specific features on elements of the RELS list, so that a word or phrase combining with this PP could link up for example to the event variable representing the *to* situation (here, 7).[3] The S node corresponding to the whole sentence similarly has a CONT

[3] For a thorough introduction to Minimal Recursion Semantics and its integration into the ERG for purposes of compositionality, see Copestake et al., 2005.

value, which encodes the semantic representation of the whole. This semantic representation can be translated from the grammar-internal, composition-ready format of Figure 2 into a grammar-external, interface representation, shown in Figure 3.

Our purpose in providing this short tour of a feature structure has been to illuminate the level of detail involved in both the grammar and the resulting representations. Of course, very large feature structures are inconvenient representations for most other kinds of processing. Most users would in fact be interested in views (or what Branco et al. (2010) call 'vistas') that present only a subset of this information, be it syntactic or semantic in nature, or blending both levels of analysis. By combining the native ERG derivation with the underlying grammar and software to deterministically rewrite or suppress parts of the HPSG sign, the Redwoods approach allows users of the treebank to dynamically parameterize and extract a range of different such views.

Figure 3 displays the grammar-independent MRS meaning representation associated with the selected analysis of (1). Similarly, Figure 4 shows a reduction into bi-lexical syntactic (top) and semantic (bottom) dependencies, as defined by Zhang and Wang (2009) and Ivanova et al. (2012). These views on the data are automatically derived and do not represent any further manual annotation effort: they are simply subsets of the highly articulated syntactico-semantic annotations that the Redwoods methodology allows us to create. Accordingly, they benefit from the same dynamic extension and refinement properties as the underlying treebank.

$$
\begin{array}{l}
\langle\, h_1, \\
\quad \left|\begin{array}{l}
h_4{:}\_\text{a\_q}(\text{BV } x_6, \text{ RSTR } h_7, \text{ BODY } h_5), \\
h_8{:}\_\text{analogous\_a\_to}(\text{ARG0 } e_9, \text{ ARG1 } x_6), h_8{:}\text{comp}(\text{ARG0 } e_{11}, \text{ ARG1 } e_9, \text{ ARG2 } \_), \\
h_8{:}\_\text{technique\_n\_1}(\text{ARG0 } x_6), \\
h_2{:}\_\text{almost\_a\_1}(\text{ARG0 } e_{12}, \text{ ARG1 } h_{13}), h_{14}{:}\_\text{impossible\_a\_for}(\text{ARG0 } e_3, \text{ ARG1 } h_{15}, \text{ ARG2 } \_), \\
h_{17}{:}\_\text{apply\_v\_to}(\text{ARG0 } e_{18}, \text{ ARG1 } \_, \text{ ARG2 } x_6, \text{ ARG3 } x_{20}), \\
h_{21}{:}\text{udef\_q}(\text{BV } x_{20}, \text{ RSTR } h_{22}, \text{ BODY } h_{23}), h_{24}{:}\_\text{other\_a\_1}(\text{ARG0 } e_{25}, \text{ ARG1 } x_{20}), \\
h_{24}{:}\_\text{crop\_n\_1}(\text{ARG0 } x_{20})
\end{array}\right| \\
\{\, h_1 =_q h_2, h_7 =_q h_8, h_{13} =_q h_{14}, h_{15} =_q h_{17}, h_{22} =_q h_{24} \,\}\rangle
\end{array}
$$

Fig. 3 Minimal Recursion Semantics for example (1).

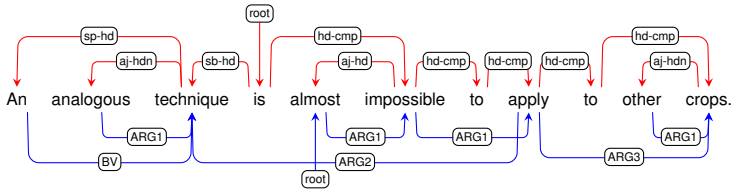

Fig. 4 Bi-lexical syntactic and semantic dependencies for (1).

The heart of the structure in Figure 3 is predicate–argument structure, encoded as a multiset of elementary predications. Each elementary prediction includes a predicate symbol, a label (or 'handle', prefixed to predicates with a colon in Figure 3),

and one or more argument positions, whose values are either logical variables or handles. MRS variable types distinguish *eventualities* (e_i), which denote states or activities, from instance variables (x_j), which typically correspond to (referential or abstract) entities. The variable x_6 appears as the argument of _technique_n_1, _analogous_a_to, and _apply_v_to. In other words, the techniques are what are analogous and what are (hypothetically, in this case) being applied. x_6 also appears as the BV ('bound variable') argument of the generalized quantifier _a_q. MRS goes beyond predicate–argument structure, however, and also provides partial information about scope. In particular, predicates such as _impossible_a_for take scopal argument positions (here ARG1) which are related via the 'handle constraints' shown in the last line (e.g. $h_{15} =_q h_{17}$) to their arguments, leaving room for quantifiers such as _a_q to take scope in different positions in the sentence. Though there are no interesting scopal effects in this example, this partially specified representation is what allows us to create one analysis of a sentence like *Every student read some book* that is consistent with either relative scoping of the quantifiers while still appropriately constraining the scope of elements like *not*.

The bi-lexical dependencies shown in Figure 4 project a subset of the syntactic and semantic information discussed so far onto a set of directed, binary relations holding exclusively between words. Here, syntactic dependency types correspond to general HPSG constructions. For example, the edge labeled HD-COMP linking *apply* to *to* in the syntactic dependencies indicates that the PP headed by *to* is functioning as a complement of the head *apply*. Similarly, semantic dependencies are obtained by reducing the MRS into a variable-free dependency graph (Oepen & Lønning, 2006), which is then further simplified to predicate–argument relations that can be captured by word-to-word dependencies (Ivanova et al., 2012). For example, the ARG2 edge that links *apply* to *technique* in the semantic dependency view indicates that the referent of *technique* plays the role of ARG2 with respect to the predication introduced by *apply* in the predicate–argument structure.

One way to conceptualize the complexity of annotations is by considering the linguistic phenomena which are represented. A 'classic' resource like the PTB, for example, avoids making quite a number of distinctions, including a clear argument vs. adjunct contrast, finer points of subcategorization, NP-internal structure, and many more. The ERG makes all of these distinctions, representing the differences in the more articulated trees as well as in the feature structures on the nodes. In many cases, the annotation decisions (discriminant choices) come down to choices along these dimensions. These distinctions represent important linguistic information in their own right, but they also support what is perhaps the most valuable layer of the ERG annotations, viz. the semantic representations. These semantic representations include semantic roles which can be seen as akin to those partially annotated in PropBank (Kingsbury & Palmer, 2002), but go much further: Every semantically contentful word in every item is reflected by one or more 'elementary predications', which are all linked together through predicate-argument structures. Furthermore, the semantic representations also reflect the semantic contribution of syntactic con-

structions via additional elementary predications.[4] Finally, they include a distinction between scopal and non-scopal argument positions and include partial information about quantifier scope.

Another way to view the complexity of the annotations in Redwoods is through the lens of the linguistic phenomena which are analyzed by the grammar. In (1) alone, we see the effects for such 'core' linguistic phenomena as the distinction between arguments and adjuncts (*almost* is an adjunct of *impossible*; *to other crops* is a argument of *apply*), subject-verb agreement; and predicative adjectives (*impossible*) and the associated (semantically empty) form of the copula (*is*). In addition, this example illustrates a more subtle linguistic phenomenon, namely *tough*-movement, wherein the object (and thus the second most promiment semantic argument) of *apply* is linked to the subject of the so-called *tough*-adjective *impossible*, while the subject (and most prominent semantic argument) of *apply* is left unexpressed. This construction and others like it are more common than might be expected, and not recovered reliably by modern stochastic parsers trained on resources like the PTB (Rimell, Clark, & Steedman, 2009; Bender, Flickinger, Oepen, & Zhang, 2011).

We argue that this level of complexity of linguistic annotation is beyond the scope of what can be developed and consistently applied if the annotations are written or even edited by hand. The methodology that we advocate allows us to create and maintain the annotations because of the way we combine the contributions of human annotators and machine assistance. The annotations are all manually designed in the sense that the work of creating the grammar in the first place entails designing the intended representations (e.g. the intended semantic representations) and then creating and constraining the rules so that those representations are made available. The annotations are further manually selected, but in a fashion that is optimized for preserving the value of every piece of manual human input, as described above.

4 Discussion: Methodological Reflections

The previous sections have presented our approach to designing and selecting annotations and argued that this approach enables the production of very detailed annotations and greatly helps in maintaining consistency in those annotations across the corpus. In this section, we look in more detail at the process of producing and updating Redwoods annotations. In particular, we describe how maintaining a treebank is critical to grammar development (§ 4.1), present some of the challenges faced by our approach and how we address them (§ 4.2), and finally discuss further strategies for improving annotation consistency (§ 4.3).

[4] An example of a syntactic construction contributing semantic information is the one that licenses determinerless or 'bare' noun phrases and inserts a quantifier elementary predication.

4.1 Grammar and Treebank

Grammar development proceeds by refining analyses of already handled phenomena and by adding analyses of new phenomena. The more phenomena a grammar analyzes, the more candidate analyses it proposes for a given sentence—that is, the more ambiguity it finds. This is because any phenomenon added to a grammar involves constraints which can be met infelicitously by substrings of sentences whose intended interpretation does not contain that phenomenon. For example, since *barks* can be a noun or a verb, any grammar that handles noun-noun compounding will find an analysis of *The dog barks.* which treats it as an NP fragment (i.e. the plural of *The dog bark*). As this example illustrates, these interactions arise even in grammars with relatively modest linguistic coverage.

Beyond the way it adds undesirable ambiguity, the inherent complexity in the interaction of constraints and rules is an important source of difficulty in grammar development. For example, constraints added to limit the applicability of newly added rules (and thus the degree of ambiguity that they introduce) can block previously available analyses of other, interacting phenomena.[5] This complexity necessitates a detailed and practical testing regime, if grammar development is to be successful: since the utility of a broad-coverage grammar depends on a healthy tension and delicate balance among the aims of efficiency in processing, robustness of coverage, and accuracy of analysis, every change to the grammar brings the real possibility of unwanted changes in the analyses licensed by the grammar for phenomena once within its demonstrated capabilities. The development and maintenance of a treebank is key to detecting any such regressions.

As a case study of the Redwoods approach to linguistic annotation, we examine the experience of grammar developers and annotators working with the ERG over the past twelve years, during a period of significant expansion of its linguistic coverage driven by several development efforts, including two commercial applications and several research projects. This expansion included a five-fold increase in the number of manual lexical entries, and a four-fold increase in the number of syntactic rules, along with the addition of unknown-word handling based on a standard part-of-speech tagger, and regular-expression-based preprocessing machinery to normalize treatment of numerals, dates, units of measure, punctuation, and the like. These enhancements dramatically improved the grammar's ability to assign linguistically viable analyses to sentences in running text across a variety of texts, including familiar corpora such as the SemCor portion of the Brown corpus and the portion of the Wall Street Journal annotated in the Penn Treebank, as well as more application-relevant corpora such as the English Wikipedia, GENIA biomedical texts, tourism brochures, and user-generated data from web blogs and news groups.

In order to preserve the grammar's success in analyzing previously studied phenomena as it extended its reach to new ones, the grammar development process

[5] Indeed the interaction of phenomena is often a primary source of evidence for or against specific analyses (see Bender, 2008; Fokkens & Bender, 2013).

came to include an essential step of comparing its current coverage to that of the previous version on each of the sentences of already-analyzed corpora. These previously confirmed sentence-analysis pairs, stored in the Redwoods Treebank, can be compared to newly produced parse forests constructed for each sentence with a revised version of the grammar, to confirm or deny that the intended analysis is still assigned by the grammar. The specialized software platform used for this version-to-version comparison of treebanked corpora is called [incr tsdb()] (Oepen & Flickinger, 1998), a competence and performance 'profiling' tool which enables the fine-grained comparison of syntactic and semantic analyses necessary for sustained grammar development.

For a given previously treebanked sentence, the comparison with a newly constructed parse forest is made by first applying the recorded binary discriminants to the new forest. Where these reduce the forest to the same tree previously recorded, the sentence is automatically confirmed as retaining the intended parse in the new version of the grammar. Where the application of the discriminants reduces the forest but results in more than one remaining analysis, it is clear that the new version of the grammar has introduced additional ambiguity which needs to be manually resolved, with the additional discriminant(s) stored in the [incr tsdb()] profile for the next development cycle. And where the old discriminants result in the rejection of all trees produced for this sentence using the new version of the grammar, it is clear that the implementation of analyses for one or more linguistic phenomena suffered damage, usually inadvertent, as the grammar was revised.[6]

In practice, the tools used for disambiguation via selection of discriminants imposed resource bounds which made it most efficient to work not with the entire parse forest for a given sentence, but rather the 500 most probable candidate analyses (as determined using a parse-ranking model trained on an earlier treebank). This 500-best limit made the storage and manipulation of the sets of analyses more tractable, even though an occasional sentence in the treebank could not be updated for a new version of the grammar because the intended analysis, while still licensed by the grammar, was no longer in the top-ranked 500 parses. Similarly, resource bounds on the parsing process itself resulted in some previously treebanked sentences failing to parse simply because the parser hit a limit using the new and more ambiguous grammar. Fortunately, these resource limit effects remain no more than a minor nuisance in the update process, together affecting less than one percent of the items in the treebank when updating from one grammar version to the next.

Much more common in the update process are those sentences for which the treebanked analysis is either no longer available, or is masked by newly added am-

[6] More precisely, the Redwoods Treebank stores for each sentence two classes of discriminants: those manually selected by the annotator, and the rest which can be inferred from the manual choices. These inferred discriminants generally add to the robustness of the annotations, offering redundant sources of disambiguation, but this redundancy can get in the way of some kinds of grammar changes. Hence the annotation update machinery includes the ability to restrict the set of old discriminants to only manually selected ones, in those instances where applying the full set of discriminants results in the rejection of all new analyses. This restriction happily often leads to successful disambiguation even given significant changes to the grammar, by ignoring inferred discriminants that were previously redundant, and are now no longer applicable.

biguity. Where a treebanked analysis has been lost, enough information has been preserved to help pinpoint the locus of change in the grammar. Since the treebank has recorded for each sentence not only the discriminants that were applied when disambiguating, but also the full derivation tree (the 'recipe' of rules that were applied to particular lexical entries), it is straightforward to 'replay' the derivation using the new grammar, to reveal to the grammarian which specific properties of words or rules have changed to block the desired analysis. Where additional ambiguity has been introduced, the annotator is presented with the new discriminants necessary to resolve it. The grammarian can review these new sources of ambiguity to see if they are intended, or if they point to the need for further tuning of the grammar to restrict the applicability of the rules involved.

The loss of treebanked items during a grammar development cycle is highly informative to the grammarian, and typically indicates the introduction of overly restrictive alterations to existing rules or lexical types while the grammarian was in pursuit of reduction of spurious ambiguity. As the storehouse of treebanked sentences grows, the treebank becomes an ever more sensitive source for detecting unintended effects of changes to the grammar, enabling the grammarian to improve grammar coverage and reduce spurious ambiguity in a largely monotonic fashion over time.

However, the benefits to the grammarian of that larger treebank come at an ever growing cost, since with each substantial grammar update cycle, some 20% of the sentences in the treebank end up requiring manual attention, even if only to resolve slight increases in ambiguity. While it typically only takes a few seconds to attend to each such sentence in an update cycle, this can add up to many hours of annotation effort to curate the existing treebank as it comes to contain tens or hundreds of thousands of sentences. Since updating of the treebank can, as noted, reveal grammar errors at any point in the update process, a cautious procedure then necessitates reparsing the full corpus and re-updating to that point, adding some additional effort to the manual annotation cost with each round of correction and updating as the grammar converges to what the grammarian intended. These preservation-based annotation costs have been sustainable as the Redwoods Treebank has grown to its current size, but this necessary and valuable updating of existing Redwoods annotations now consumes more than half of the effort required when making substantive annual expansions of coverage for the ERG. With the recent addition of the WSJ portion of Redwoods, effectively doubling its size, the maintenance cost for the next update is likely to increase proportionately, and it is clear that our tools and methods for treebanking will need to evolve toward better automation and reduced human effort.

4.2 Challenges for Treebanking New Corpora

Since the construction of a Redwoods treebank centers on manual disambiguation among the candidate analyses licensed for each sentence by the chosen grammar,

consistency in the selection of discriminants distinguishing the analyses is essential, but challenging. Many of the contrasts presented by the grammar for a given sentence correspond well to an annotator's intuitions about its expected structure or meaning, but some residual ambiguity can be difficult to resolve, either because the alternatives appear to be semantically neutral, or because the choice requires specialist knowledge of the domain.

For some linguistic constructions, the grammar may present multiple candidate analyses each of which is well motivated given the principles of the syntactic theory, but which do not differ semantically. For example, the attachment of a sentence-final subordinate clause in English is proposed by the ERG either as a modifier of the verb phrase or of the full sentence. Making both analyses available is motivated by the interaction with the analysis of coordination. Thus, the sentence in (2a) will include two semantically identical analyses reflecting the two possible attachments, motivated by the two variants in (2b,c), where in the first case each VP conjunct contains a clausal modifier, while in the second, the conditional clause can scope over the conjunction of the two full sentences.

(2) a. They will take a cab if the plane arrives late.

b. They will take a cab if it's late and ride the bus if it's on time.

c. They will take a cab and we'll call our friends if it's late.

Since the grammar must allow the conditional clause to attach either to a VP or to an S, the first example above will include analyses with each of these two attachments, but the meaning representation (the MRS) is the same. In such cases, the annotator will have to make a discriminant choice which is determined not by intuition but by convention, based on a set of annotation guidelines.

In other constructions, ambiguity may correspond to semantic distinctions that are formally clear but irrelevant in the given domain, again driving the annotator to make discriminant choices based on annotation guidelines rather than on linguistic or domain knowledge. For example, the ERG assigns binary structures to compound nouns, presenting the annotator with two distinct bracketings for a phrase such as *airline reservation counter*, where it is normally irrelevant whether it is a counter for making airline reservations, or a reservation counter operated by an airline. Similarly, attachment of prepositional phrases is sometimes not semantically significant, as in the following example:

(3) They reserved a room for Abrams.

Here again it may not matter whether there was a reservation action that involved a room for Abrams, or whether a room got reserved as a service to Abrams. Annotation guidelines to ensure consistency in these instances are more difficult to apply, since annotators may not agree on when a semantic distinction is irrelevant.

A third class of annotation difficulties arises when the resolution of an ambiguity requires highly specialized domain knowledge. For example, the following sentence

from the GENIA corpus includes the phrase *their natural ligands, glucocorticosteroids and catecholamines*, which might be either an apposition of *glucocorticosteroids and catecholamines* as types of ligands, or instead a three-part coordination of nominal phrases.

(4) When occupied by their natural ligands, glucocorticosteroids and catecholamines, these receptors have a role in modulating T-cell function during stress.

Here the disambiguation is semantically significant, but the discriminant choice might have to be deferred until the necessary domain knowledge can be obtained. Such collaboration between the linguistically informed annotator and the domain specialist can significantly increase the time needed to construct a treebank which accurately reflects the relevant semantic distinctions correlated with syntactic structures. An alternative method, applied by MacKinlay, Dridan, Flickinger, Oepen, and Baldwin (2011), adopts an annotation convention to assign a default bracketed structure to such phrases where specialist knowledge would be required, ideally further marking such items for later refinement. Either way, once the domain expert's knowledge is incorporated into the annotation decisions, this information is carried forward, without further effort, in future updates of the treebank.

4.3 Improved Consistency of Annotation in the Existing Treebank

While the particular sources of ambiguity discussed above present challenges for consistency in annotation, they can be addressed in large part through the adoption and documentation of conventions for discriminant choice. However, the existing Redwoods Treebank contains other inconsistencies which have several sources, including human error, incompleteness of the annotation guidelines, and the complexity of exhaustive annotation for every constituent, particularly multi-token named entities. Manual review and correction can reduce the number of annotation errors over time, but better methods for automatic detection of candidate errors may enable further refinement of the resource, as can revisions to the grammar to remove remaining spurious ambiguity.

For some phenomena, particularly multi-token named entities such as *New York Stock Exchange* or *the Wall Street Journal*, detailed annotation conventions can be augmented with software support for defining and applying corpus-specific labeled bracketing defaults during parsing, to ensure consistency for the most frequently occurring such named entities in a corpus. The ERG includes support for the preservation of externally supplied constituent bracketing when parsing, which enables the grammarian to define such multi-token named entity bracketings.

More vexing are the remaining sources of spurious ambiguity in the grammar, presenting variant analyses which are not clearly motivated linguistically, but instead result either from complex interactions among well-motivated constraints, or from contrasts that have become less well-defined as the grammar has evolved. An

example of the latter appears in our running example, where *almost impossible* is analyzed by the ERG both as a modifier–head structure and as a specifier–head structure. Adjectives in English do impose some clear requirements on the degree phrases that precede them, so the contrast between *very/*much tall* and *much/*very taller* is ensured via constraints by adjective heads on their specifiers. However, adjectives can also be preceded by many of the same elements that are treated as ordinary modifiers when combining with verb phrases, as in *obviously impossible* or *often impossible*, so the grammar also licenses adjectives as heads of such modifier-head phrases. Then for an element like *almost*, which expresses a constraint on degree but also appears as a verbal modifier, both structures are admitted for *almost impossible*, presenting the annotator with a non-intuitive discriminant choice. Minimizing such ambiguity in the grammar would of course improve the consistency and reduce the cost of annotation, but when the necessary refinements involve analyses of core phenomena, changes can have subtle consequences that may be detectable only with the aid of a substantial existing treebank.

4.4 Summary

This section has presented some reflections on the methdology of the Redwoods Treebank. The central ideas of the methodology—encoding the design of the annotations in a machine-readable grammar and using dynamic discrinimant-based treebanking to choose among analyses provided by the grammar—support *scalability*, both in complexity of annotations and in the size and genre diversity of the treebank. A result of the approach which was not apparent a priori is the synergistic development of grammar and treebank, where effort on one informs and improves the other. Even with the grammar encoding the annotation design, there still remain questions of consistency to address, especially across genres, and room for further software-based solutions to these issues. In the next section, we situate our methodology with respect to related work.

5 Neighborhood: Related Work

In the above, we argue that grammar-based dynamic annotation is a viable approach to the creation of large, multi-layered, and precise treebanks. Existing such resources like the Prague Dependency Treebank (Hajič, 1998) or the ecosystem of distinct but interoperable annotation layers over the PTB (and more recently the OntoNotes collection; Hovy, Marcus, Palmer, Ramshaw, & Weischedel, 2006) suggest that grammar-based annotation is far from being the only possible path towards

rich annotation at scale.[7] But these resources are scarce and mostly static over time: in part for both technical and cultural reasons, there is no mechanism for correcting known deficiencies in PTB syntactic analyses, for example. More importantly, we conjecture that grammar-based annotation can be far more cost-efficient and lead to greater consistency; in other words, this approach exhibits better scalability. In the following, we survey some closely related initiatives.

As we observed in §2 above, many of the foundational ideas behind the Redwoods approach are due to Carter (1997). With the primary goal of creating domain-specific training data for the stochastic disambiguation component in the Core Language Engine (CLE; Alshawi, 1992), he developed the TreeBanker, a discriminant-driven graphical tool for selecting the preferred analysis from the CLE parse forest. Reflecting different levels of analysis in the underlying grammar, the TreeBanker had support for disambiguation in terms of both syntactic and semantic properties, with special emphasis on foregrounding discriminants that are expected to be easy to judge by non-experts, for example attachment contrasts for prepositional phrase modifiers. The original description by Carter (1997) mentions briefly the option of 'merging' existing disambiguation decisions into the discriminant space resulting from parsing the same input after extending the grammar for coverage, but there is no discussion of the specific design and strategy choices for this operation (see §4.1 above). For low- to medium-complexity sentences (in the venerable ATIS flight reservation domain), Carter (1997) reports disambiguation rates of between 50 and 170 sentences per hour, which would seem to compare favorably to the rate of some 2,000 sentences per week reported by Oepen et al. (2004) for the earlier Redwoods years. However, it appears the TreeBanker has never been applied to the construction of large-scale treebanks, actively maintaining and refining annotations over a larger volume of naturally occurring text over time.

At about the same time as the creation of the First Growth of the Redwoods Treebank, van der Beek, Bouma, Malouf, and van Noord (2002) at the University of Groningen worked towards the creation of the Alpino Dependency Treebank for Dutch, which instantiates the same abstract setup. The treebank is constructed by manual, discriminant-based disambiguation among the set of analyses produced by a broad-coverage, computational grammar of Dutch (Bouma, van Noord, & Malouf, 2001).[8] Despite much abstract similarity, there are some important differences. Firstly, the Alpino Treebank is exclusively comprised of syntactic dependency structures, i.e. a single layer of analysis, which eliminates much of the flexibility in extracting dynamic views on linguistic structure that the Redwoods architecture affords.[9] Secondly, and maybe more importantly, the Groningen initiative allows man-

[7] And, naturally, the contrast of approaches is not at all black-and-white, as there are bound to be elements of data preparation or guiding annotators through automated analysis (e.g. tagging and syntactic parsing) in most contemporary annotation work.

[8] The contemporaneous development of two initiatives in grammar-based treebanking is not entirely coincidental, as the original Redwoods tree selection tool was developed by Rob Malouf, prior to his joining the Alpino team at Groningen.

[9] More recent work at Groningen has focused on annotated resources that combine syntactic and semantic representations, this time for English, in the form of the Groningen Meaning Bank (Basile,

ual correction (post-editing) of dependency structures constructed by the grammar. Thus, it makes the assumption that syntactic analyses, once corrected and recorded in the treebank, are correct and do not change over time (or as an effect of grammar evolution); accordingly, disambiguating decisions made by annotators are *not* recorded in the treebank, nor does the project expect to dynamically update annotations with future revisions of the underlying grammar.

Another related approach is the work reported by Dipper (2000) at the University of Stuttgart, essentially the application of a broad-coverage Lexical-Functional Grammar (LFG) implementation for German to constructing tectogrammatical structures for the German 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: target representations are mono-stratal and the connection to the original LFG analyses and basic properties used in disambiguation are not preserved in the treebank.

The Redwoods methodology and tools have been applied to other languages for which HPSG implementations of sufficient coverage exist, and generalized to support disambiguation in terms of 'classic' syntactic discriminants as well as through semantic ones, i.e. a basic contrast in predicate–argument structure (Oepen & Lønning, 2006). Languages for which Redwoods-like treebanking initiatives are underway include Japanese (Bond et al., 2004), Portuguese (Branco et al., 2010), Spanish (Marimon et al., 2012), and recently Bulgarian (Flickinger, Kordoni, et al., 2012). There are important differences between these initiatives in scope, choice of text types to annotate, and nature of discriminants used, but they all embrace the same development cycle as Redwoods, integrating tightly the incremental refinement of the annotation design, through grammar adaptation, with the sustained maintenance of an ever growing collection of annotated text.

In more recent work, the same basic approach has been successfully adapted to discriminant-based, dynamic treebanking with large LFG implementations by Rosén, Meurer, and De Smedt (2007). For Norwegian in particular, an ongoing large-scale initiative at the University of Bergen is working towards a 500,000-word collection of running text that is paired with full, manually selected and validated LFG analyses. There are important linguistic and technical differences, again, but the in-depth experience report of Losnegaard et al. (2012) suggests that this initiative has opted for an even tighter coupling of grammar refinement and treebank updates, or at least for more frequent iterations of the basic bi-directional feedback loop sketched above.

Bos, Evang, & Venhuizen, 2012). This work, however, does not build on either a precision handcrafted grammar or a discriminant-based treebanking strategy, so it is of less direct relevance here.

6 Outlook: Further Challenges

While the Redwoods methodology has much to recommend it for the construction and steady enhancement of ever larger linguistically annotated corpora, several challenges remain as opportunities for improvements in the tools and in the annotations. Some of the shortcomings may be addressed soon by ongoing work, while others are likely to keep researchers engaged for some time to come.

Among the near-term improvement opportunities is the existing practical limit in the annotation tool chain of just the 500 most likely analyses for a given sentence to be treebanked. Since all of the available parsers for grammars like the ERG can construct a compact packed forest of all of the analyses licensed by the grammar for a sentence, it would be better to treebank the full parse forest rather than just the top 500, for reasons given in § 4.1. A utility which supports this more comprehensive annotation has been developed recently (Packard, in preparation) and should be ready for use in the next release cycle for the ERG, bringing greater stability to the resulting treebank.

Another challenge for this grammar-centric method of annotation is that a grammar implementing a linguistic theory will fail to provide a full correct analysis of some sentences in a corpus of any size, either because a sentence instantiates a linguistic construction not yet adequately studied in the theory, or because the grammar does not successfully implement the intended treatment of some construction. Given the current state of the ERG, 5-10% of the sentences in most of the corpora studied so far fail to receive any analysis at all from the ERG, and another 5-10% receive some analyses but not correct ones (Flickinger, 2011). While this gap in linguistic coverage has been shrinking over the years, it will not soon disappear, thus leaving some portion of a typical corpus to be annotated by other means. One recent and promising approach by Zhang & Krieger, 2011 uses a probabilistic CFG trained on a large corpus of ERG-parsed text to produce approximately correct syntactic analyses, which can be used as the basis for computing an approximate MRS for each sentence that is lacking annotation in the treebank for a given corpus.

A third challenge in the Redwoods approach involves the lack in the annotations of aspects of linguistic content that are desirable but not yet deriveable given the existing grammars and tools. Fine-grained word senses, anaphoric co-reference within and across sentences, information structure, and discourse relations are examples of annotation elements that are not yet included in the Redwoods Treebank, but might be added in the foreseeable future. As noted above, it is a strength of this approach that refinements or enrichments to the annotations can be added inexpensively and consistently to already annotated text by updating the grammar to produce the new annotations. An example involves the analysis of appositives, such as (5):

(5) Abrams, the chairman of the board, arrived.

The current semantic analysis implemented in the ERG relates the indices of the two NPs (*Abrams* and *the chairman of the board*) via a two place relation called appos. This relation is introduced by the syntactic rule that licenses the juxtaposition of the two NPs. The semantic analysis of this construction is a topic of current research.

One candidate alternative analysis involves an addition to the semantic structure called ICONS, a multiset of 'individual constraints' relating semantic variables. On this proposal, the identity of reference between the two NPs in an appositive construction would be represented as an ICONS constraint. This is a particularly simple case of an update of annotations, since the exact same syntactic configuration is involved; once the semantic constraints on the syntactic construction are updated in the grammar, reparsing the corpus and rerunning the discriminant selections will result in a disambiguated treebank with the new annotations.

Other types of enrichments of the semantic structures require different approaches, but we argue that these can still be achieved in a manner that maximizes the value of any manual annotation. A first example is annotations capturing information structure. A representation of information structural constraints (e.g. the assigment of parts of the semantic representation to topic, focus, or background) using ICONS has been proposed by Song and Bender (2012), and there are several rules in the grammar which can be updated to reflect the partial constraints on information structure that constructions like *it*-clefts and fronting provide. As above, this would immediately lead to enrichment of the annotations in the treebank without further manual work.

However, English morphosyntax provides only very little information about roles such as topic and focus. Most sentences in isolation are highly ambiguous at this level. Since there is nothing in the syntax to disambiguate further, we argue that having the grammar enumerate all possibilities is inefficient—it increases processing time and complicates the parse selection process when the grammar is used online for analysis. We thus propose instead a pipeline approach, where additional candidate annotations such as fully specified annotation for focus/topic, coreference chains or fine-grained word sense distinctions, are provided by a separate processor over the gold syntactico-semantic annotations selected in the treebank. A similar discriminant-based approach can be deployed over these options, reducing the set of full analyses for each sentence to a set of binary choices for the annotator to consider, which can similarly be rerun after a re-processing pass. Though we do not yet have such a pipeline set up, we emphasize here that the semantic annotations are ready to be extended in this fashion, for multiple purposes: ICONS can be used to represent coreference chains as well as information structure, and the semantic predicates in MRS can be mapped, one-to-many, to e.g. WordNet senses (Fujita, Bond, Tanaka, & Oepen, 2010; Pozen, 2013).

Once we open the possibility of adding annotations through post-processing (and then applying a similar discriminant-based approach to selecting among them), we face the question of whether other annotation decisions that are currently handled within the grammar might be better treated in a similar fashion. Some candidate examples here include PP attachment ambiguities and the internal bracketing of noun-noun compounds. While the syntax provides a range of possibilities, there are relatively few dependencies between these decisions and anything else in the grammar: In *Abrams went to the airline reservation counter.* nothing else in the sentence provides any constraints on the whether *reservation* combines first with *airline* or

counter. Similarly, in *Browne reserved a room for Abrams.* the PP attachment decision is independent of all other syntactic disambiguation steps.

However, internal to longer chains of either type of ambiguity there are interactions. The bracketings in (6a,c), for example, preclude the bracketings in (6b,d):

(6) a. Abrams went to the airline [ticket reservation] counter.
b. Abrams went to the [airline ticket] reservation counter.
c. Browne reserved [a room for Abrams in Reykjavik].
d. Browne [[reserved a room for Abrams] in Reykjavik].

The syntactic structures that we assign automatically calculate these dependencies for us, and so while it is appealing to underspecify PP attachment and noun-noun-compound bracketing, our current approach disambiguates these in the syntax.

Conversely, there are potential sources of syntactic ambiguity that the grammar rules out from the start, since they never lead to differences in semantic representations. A case in point is the order of attachment of intersective modifers. Since some appear pre-nominally and some post-nominally, there is a choice as to which to attach first which is not constrained by linear order in the string. The ERG currently implements a blanket heuristic of attaching post-modifiers before pre-modifiers.

Over time, we can expect to see continued enhancements not only in the consistency of Redwoods annotations, but also in their density and variety, including layers of linguistic analysis produced not just by the grammars and parsers, but by other utilities that can integrate their contributions with the representations currently available.

7 Conclusion

We began this chapter with a thought experiment focused on issues of scale—scaling linguistic annotations to very large, genre-diverse corpora and scaling linguistic annotations in their complexity and comprehensiveness. We have argued that working towards such large-scale ambitions requires careful management of human effort and preservation of the results of any manual labor. The methodology that we describe here answers these requirements: linguistic analytical effort is focused on two main activities, viz. the development of a linguistically-motivated, precise and broad-coverage grammar and the disambiguation of the set of analyses provided by the grammar via 'discriminants'. This methodology supports the development and consistent deployment of annotations with much greater complexity than could be managed without such machine assistance. Furthermore, it supports the incremental improvement and elaboration of those annotations, as the underlying corpus can be reparsed whenever the grammar is updated to refine or extend the annotations and the discriminant choices rerun. With our thought experiment, we deliberately invoked an unachievable ideal case in order to broaden the range of possibilities under consideration. As discussed above, there remain many areas for future work,

both problems to solve within the purview of the current annotation domains as well as directions for extensions of the annotations beyond those which are closely tied to morphosyntax; nonetheless, we contend that our methodology represents a substantial step towards comprehensive, maintainable, and scalable annotation.

References

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

Adolphs, P., Oepen, S., Callmeier, U., Crysmann, B., Flickinger, D., & Kiefer, B. (2008). Some fine points of hybrid natural language parsing. In *Proceedings of the 6th International Conference on Language Resources and Evaluation.* Marrakech, Morocco.

Alshawi, H. (Ed.). (1992). *The Core Language Engine*. Cambridge, MA, USA: MIT Press.

Basile, V., Bos, J., Evang, K., & Venhuizen, N. (2012). UGroningen. Negation detection with Discourse Representation Structures. In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics* (p. 301–309). Montréal, Canada.

van der Beek, L., Bouma, G., Malouf, R., & van Noord, G. (2002). The Alpino dependency treebank. In M. Theune, A. Nijholt, & H. Hondorp (Eds.), *Computational linguistics in the Netherlands 2001. selected papers from the twelfth CLIN meeting.* Amsterdam, The Netherlands: Rodopi.

Bender, E. M. (2008). Grammar engineering for linguistic hypothesis testing. In N. Gaylord, A. Palmer, & E. Ponvert (Eds.), *Proceedings of the Texas Linguistics Society X Conference. Computational linguistics for less-studied languages* (p. 16–36). Stanford, USA: CSLI Publications.

Bender, E. M., Flickinger, D., Oepen, S., & Zhang, Y. (2011). Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing* (p. 397–408). Edinburgh, Scotland, UK.

Bond, F., Fujita, S., Hashimoto, C., Kasahara, K., Nariyama, S., Nichols, E., … Amano, S. (2004). The Hinoki Treebank. A treebank for text understanding. In *Proceedings of the 1st International Joint Conference on Natural Language Processing* (p. 158–167). Hainan Island, China.

Bouma, G., van Noord, G., & 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* (p. 45–59). Amsterdam, The Netherlands: Rodopi.

Branco, A., Costa, F., Silva, J., Silveira, S., Castro, S., Avelãs, M., … Graça, J. (2010). Developing a deep linguistic databank supporting a collection of treebanks. The CINTIL DeepGramBank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation.* Valletta, Malta.

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* (p. 9–15). Madrid, Spain.

Copestake, A., Flickinger, D., Pollard, C., & Sag, I. A. (2005). Minimal Recursion Semantics. An introduction. *Research on Language and Computation*, *3*(4), 281–332.

Dipper, S. (2000). Grammar-based corpus annotation. In *Proceedings of the Workshop on Linguistically Interpreted Corpora* (p. 56–64). Luxembourg, Luxembourg.

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

Flickinger, D. (2011). Accuracy vs. robustness in grammar engineering. In E. M. Bender & J. E. Arnold (Eds.), *Language from a cognitive perspective: Grammar, usage, and processing* (pp. 31–50). Stanford: CSLI Publications.

Flickinger, D., Kordoni, V., Zhang, Y., Branco, A., Simov, K., Osenova, P., ... Castro, S. (2012). ParDeepBank. Multiple parallel deep treebanking. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories* (p. 97–108). Lisbon, Portugal: Edições Colibri.

Flickinger, D., Oepen, S., & Ytrestøl, G. (2010). WikiWoods. Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation.* Valletta, Malta.

Flickinger, D., Zhang, Y., & Kordoni, V. (2012). DeepBank. A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories* (p. 85–96). Lisbon, Portugal: Edições Colibri.

Fokkens, A., & Bender, E. M. (2013). Time travel in grammar engineering. Using a metagrammar to broaden the search space. In D. Duchier & Y. Parmentier (Eds.), *Proceedings of the ESSLLI Workshop on High-Level Methodologies in Grammar Engineering* (p. 105–116). Düsseldorf, Germany.

Fujita, S., Bond, F., Tanaka, T., & Oepen, S. (2010). Exploiting semantic information for HPSG parse selection. *Research on Language and Computation*, *8*(1), 1–22.

Gawron, J. M., King, J., Lamping, J., Loebner, E., Paulson, E. A., Pullum, G. K., ... Wasow, T. (1982). Processing English with a Generalized Phrase Structure Grammar. In *Proceedings of the 20th Meeting of the Association for Computational Linguistics* (p. 74–81). Toronto, Ontario, Canada.

Hajič, J. (1998). Building a syntactically annotated corpus. The Prague Dependency Treebank. In *Issues of valency and meaning* (p. 106–132). Prague, Czech Republic: Karolinum.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). Ontonotes: The 90% solution. In *Proceedings of Human Language Technologies: The 2006 Annual Conference of the North American Chapter of the Association for Computational Linguistics, companion volume: Short papers* (p. 57–60). New York City, USA.

Ivanova, A., Oepen, S., Øvrelid, L., & Flickinger, D. (2012). Who did what to

whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop* (p. 2–11). Jeju, Republic of Korea.

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* (p. 535–541). College Park, USA.

Kingsbury, P., & Palmer, M. (2002). From TreeBank to PropBank. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation* (p. 1989–1993). Las Palmas, Spain.

Losnegaard, G. S., Lyse, G. I., Thunes, M., Rosén, V., Smedt, K. D., Dyvik, H., & Meurer, P. (2012). What we have learned from Sofie. Extending lexical and grammatical coverage in an LFG parsebank. In *Proceedings of the META-RESEARCH Workshop on Advanced Treebanking at LREC2012* (p. 69–76). Istanbul, Turkey.

MacKinlay, A., Dridan, R., Flickinger, D., Oepen, S., & Baldwin, T. (2011). Using external treebanks to filter parse forests for parse selection and treebanking. In *Proceedings of the 2011 International Joint Conference on Natural Language Processing* (p. 246–254). Chiang Mai, Thailand.

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

Marimon, M., Fisas, B., Bel, N., Villegas, M., Vivaldi, J., Torner, S., ... Villegas, M. (2012). The IULA Treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation* (p. 1920–1926). Istanbul, Turkey.

Oepen, S., Flickinger, D., Toutanova, K., & Manning, C. D. (2004). LinGO Redwoods. A rich and dynamic treebank for HPSG. *Research on Language and Computation*, *2*(4), 575–596.

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

Oepen, S., & Lønning, J. T. (2006). Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (p. 1250–1255). Genoa, Italy.

Packard, W. (in preparation). *Full forest treebanking*. Unpublished master's thesis, University of Washington.

Pollard, C., & Sag, I. A. (1987). *Information-based syntax and semantics. Volume 1: Fundamentals*. Stanford, USA: CSLI Publications.

Pollard, C., & Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar*. Chicago, USA: The University of Chicago Press.

Pozen, Z. (2013). *Using lexical and compositional semantics to improve HPSG parse selection*. Unpublished master's thesis, University of Washington.

Rimell, L., Clark, S., & Steedman, M. (2009). Unbounded dependency recovery for parser evaluation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (p. 813–821). Singapore.

Rosén, V., Meurer, P., & De Smedt, K. (2007). Designing and implementing discriminants for LFG grammars. In M. Butt & T. H. King (Eds.), *Proceedings of the 12th International LFG Conference.* Stanford, USA.

Song, S., & Bender, E. M. (2012). Individual constraints for information structure. In S. Müller (Ed.), *Proceedings of the 19th International Conference on Head-Driven Phrase Structure Grammar* (p. 330–348). Stanford, CA, USA: CSLI Publications.

Zhang, Y., & Krieger, H.-U. (2011). Large-scale corpus-driven PCFG approximation of an HPSG. In *Proceedings of the 12th International Conference on Parsing Technologies* (p. 198–208). Dublin, Ireland.

Zhang, Y., & Wang, R. (2009). Cross-domain dependency parsing using a deep linguistic grammar. In *Proceedings of the 47th Meeting of the Association for Computational Linguistics* (p. 378–386). Suntec, Singapore.