Dimensions of Deep Grammar Validation

Dan Flickinger
Oslo, Saarland, and Stanford Universities
danf@csli.stanford.edu

Abstract

In order to arrive at a more disciplined approach to the sustained development of linguistically rich grammars, I present a methodology for grammar validation, identifying principal dimensions of the task, and illustrating the application of the method for one release cycle of the open-source English Resource Grammar.

1 Introduction

Broad-coverage grammars for natural language processing which encode rich linguistic description are resources which develop relatively slowly, through many iterations of modification and testing, and ideally through exposure to the demands of a variety of NLP tasks. As with most large software components, such grammars are designed to exhibit a complexity of behavior which presents serious challenges for quality assurance from one release of a grammar to the next release. There is by now a substantial literature on many aspects of the evaluation of NLP software, including grammars, much of it focused around 'black box' or functional evaluation within some specific task domain(Hirschman and Thompson, 1998), (Sparck Jones and Galliers, 1998). Methods and tools have also been developed for 'glass box' or structural evaluation (EAGLES, 1996), with one line of work analyzing the internal formal coherence of such deep grammars, for example to flag inconsistencies or identify unused rules or constraints in the grammar code (Broeker, 2000), (Barr and Siefring, 2004), and another line of work using paraphrase generation to illuminate properties of a bi-directional grammar not easily detected when parsing (Dymetman and Isabelle, 1988). And finally, there has been work on developing a methodology of sustained grammar development, whereby these labor-intensive, long-lived resources undergo periodic modification intended to improve corpus coverage, or processing efficiency, or linguistic precision, or more typically all three at once (Oepen and Flickinger, 1998). Such a methodology must incorporate a set of disciplined procedures for validation of the resulting grammar, ensuring that it meets the expectations of its engineers, and explicitly communicates the interface specifications for its use in applications.

In this talk I describe an instantiation of a method for natural language grammar validation (cf. (Barr and Klavans, 2001)) to identify and motivate the multiple dimensions of the procedure, and to illustrate how many of the the tools and techniques proposed in the NLP evaluation literature are being exploited by engineers of large deep grammars. For concreteness, I present the method used in maintaining and extending the English Resource Grammar (ERG (Flickinger, 2000), a semantically precise, broad-coverage Head-driven Phrase Structure Grammar (HPSG) implementation used for both parsing and generation in several NLP applications, being developed within the Deep Linguistic Processing with HPSG Initiative (DELPH-IN: www.delph-in.net).

At the heart of this method is the use of a growing set of test suites of two kinds, together with a sophisticated grammar profiling tool, [incr tsdb()], which collects and preserves competence and performance data on these test suites. Some of the test suites consist of sets of hand-built example sentences and ungrammatical strings illustrating core linguistic phenomena, while the other class of test suites contain naturally occurring text items drawn from corpora, intended to be representative of the language data expected for a particular domain or task. For each of these test suites, human annotators have identified the intended analysis for each item out of the exhaustive set of analyses that the grammar supplied (up to the limits of available computational resources), thus providing 'gold standard' treebanks against which a modified version of the grammar can be mea-

Proceedings of the Australasian Language Technology Workshop 2005, pages 1–3,
Sydney, Australia, December 2005.

sured. Employing test suites from a variety of domains and tasks not only provides a basis for 'snapshot' profiles that enable a historical view of the grammar's development, but more importantly protects against overfitting of the grammar to just the current domain's data.

The principal dimensions in this grammar validation stand in some interesting tension with each other, since in each incarnation the grammar seeks to maximize (1) **robustness** in the range of phenomena and sheer quantity of data it processes; (2) **precision** linguistically in its mappings between strings and semantic representations; (3) **efficiency** in its consumption of CPU and memory resources; and (4) **stability** in the mappings it previously assigned to the items in those test suites, to minimize adaptation costs for its customer base, and to minimize the cost of updating the gold standard treebanks. For each of these four dimensions, I will present tools and techniques used to evaluate the relevant properties of the grammar, and illustrate the tensions among these dimensions with examples from the existing inventory of test suites.

Other dimensions that enter into this validation method arise from the heterogeneity of contexts in which the grammar can be used, including (5) multiple NLP **processing systems** (including at least the LKB (Copestake, 2002) and PET (Callmeier, 2000)); (6) **bi-directionality** of processing with attendant demands on each of the principal dimensions above for both parsing and generation; (7) multiple **configurations** of the grammar, including variants of preprocessing, unknown-word handling, root (start symbol) constraints, chart packing (for either parsing or generation, or both), and storage of the lexicon as text file vs relational database; (8) stochastic **parse/realization selection** or disambiguation (Oepen et al., 2004), particularly for highly ambiguous items where resource limitations become a crucial factor; and of course (9) the demands of **multiple applications**, currently including generation for Norwegian-to-English machine translation in the LOGON project (Lønning et al., 2004), extraction of ontological relationships by parsing dictionary definitions, and robust interpretation of transcribed conversations via enriched annotations of rhetorical relations.

Finally, for the grammar to be of use to application developers without an overly intimate knowledge of it, each release must be accompanied by (10) external **interface specifications** whose currency must be validated. The content of these specifications is in part automatically generated from the implemented representations of lexical entries, lexical types and grammar rules, and in part manually maintained. Principal among these specifications is the SEM-I (semantic interface) (Flickinger et al., 2005), an exhaustive listing of each lexical semantic predicate and its salient properties, which should precisely determine the observed variation in the elementary predications within the Minimal Recursion Semantics (MRS (Copestake et al., 2006)) representations that the grammar produces or accepts as input. A second essential specification provides the set of available lexical entry types, particularly those for open-class words which will inevitably need to be added for each new application, whether automatically guessed via part-of-speech tagging or entered into the lexicon manually.

In the talk, I will provide a detailed step-by-step tour of this method of validation as applied to the ERG for one typical grammar update, noting along the way how each of the dimensions identified above comes into play, often more than once in the process, and illustrating the interactions among several pairs of these dimensions in arriving at what must always be a compromise resolution of the tensions inherent in grammar engineering.

2 Acknowledgements

This work draws from research on the ERG that has been supported by several funding sources to the LinGO project at CSLI, Stanford University, over many years, including two grants from the German BMBF as part of the Verbmobil project, two National Science Foundation grants (IRI-9612682 and BCS-0094638), two grants from NTT Communication Science Laboratories, and one grant from Scottish Enterprise through the Stanford- Edinburgh Link. In addition I am grateful for current support from the LOGON project at Oslo University, and from the Computational Linguistics department at Saarland University. For the issues discussed in this talk, I am especially grateful for my recent collaborations with Stephan Oepen, Ann Copestake, Francis Bond, and Tim Baldwin.

References

Valerie Barr and Judith Klavans. 2001. Verification and validation of language processing systems: Is it evaluation? *Proceedings of the Workshop on Evaluation Methodologies for Language and Dialogue Systems, ACL2001.*

Valerie Barr and Ellen Siefring. 2004. Verification of lexicalized tree adjoining grammars. *Online Proceedings of the Seventh International Workshop on TAG and Related Formalisms.*

Norbert Broeker. 2000. The use of instrumentation in grammar engineering. *Proceedings of COLING 2000.*

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

Ann Copestake, Dan Flickinger, Ivan A. Sag, and Carl Pollard. 2006. Minimal Recursion Semantics. An introduction. *Research on Language and Computation.* (to appear).

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars.* CSLI Publications, Stanford, CA.

Marc Dymetman and Pierre Isabelle. 1988. Reversible logic grammars for machine translation. *Proceedings of the 2nd International Conference on Theoretical and Methodological Issues in the Machine Translation of Natural Languages.*

EAGLES. 1996. *Evaluation of Natural Language Processing Systems. Final Report EAG-EWG-PR.2.* EU Report.

Dan Flickinger, Jan Tore Lønning, Helge Dyvik, Stephan Oepen, and Francis Bond. 2005. SEM-I rational MT: Enriching deep grammars with a semantic interface for scalable machine translation. In *Machine Translation Summit X*, Phuket. (to appear).

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

Lynette Hirschman and Henry Thompson. 1998. Overview of evaluation in speech and natural language processing. In G. Varile and A. Zampolli, editors, *Survey of the State of the Art in Human Language Technology.* Cambridge University Press, New York.

Jan Tore Lønning, Stephan Oepen, Dorothee Beermann, Lars Hellan, John Carroll, Helge Dyvik, Dan Flickinger, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård, Victoria Rosén, and Erik Velldal. 2004. LOGON. A Norwegian MT effort. In *Proceedings of the Workshop in Recent Advances in Scandinavian Machine Translation*, Uppsala, Sweden.

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

Stephan Oepen, Daniel Flickinger, Kristina Toutanova, and Christopher D. Manning. 2004. LinGO Redwoods. A rich and dynamic treebank for HPSG. *Journal of Language and Computation.*

Karen Sparck Jones and Julia Galliers. 1998. *Evaluating Natural Language Processing Systems.* Springer-Verlag, Berlin.