Scalable Grammar Software for Computational Linguists

Building, Testing, and Running Grammars Large and Small

— Software Demonstration at ACL 2000 —

Ulrich Callmeier[♣], **Ann Copestake**[♠], **Dan Flickinger**[♡], **Rob Malouf**[♢], **and Stephan Oepen**[♣]

[♣]Computational Linguistics
Saarland University
Saarbrücken (Germany)
{uc|oe}@coli.uni-sb.de

[♠]Cambridge University
and CSLI Stanford
Cambridge (UK)
aac@cl.cam.ac.uk

[♡]CSLI Stanford and
YY Software Corporation
Palo Alto (USA)
dan@csli.stanford.edu

[♢]Alfa Informatica
University of Groningen
Groningen (NL)
malouf@let.rug.nl

Keywords

- Open-source unification-based grammar development environment;
- broad-coverage declarative grammars for several languages;
- efficient unification, parsing, and generation algorithms;
- systematic profiling and cross-platform comparison; and
- collaborative, multi-site development of NLP resources and software.

Content

We will demonstrate a suite of integrated software tools and resources for research and teaching in computational linguistics which are publicly available both separately and together, including (i) the LKB grammar development system, (ii) the broad-coverage LinGO English Resource Grammar, together with other comprehensive Verb*mobil* grammars of Japanese and German, (iii) a set of smaller teaching grammars for English and Spanish, (iv) the [incr tsdb()] profiling environment, and (v) the PET software for efficient parsing and flexible experimentation.

These tools and resources have been developed and tested over a ten-year period, with an intensive three-site collaborative effort during the past five years between the German National Research Centre of AI (DFKI) and the Computational Linguistics department in Saarbrücken, CSLI at Stanford University, and the University of Tokyo, aimed at achieving significant improvements in efficiency for large-scale unification-based grammars using typed feature structures. Research applications include spoken language machine translation in Verb*mobil*, generation for a speech prosthesis, and automated email response, under development for commercial use. Teaching uses have included undergraduate and graduate syntax courses at several sites, grammar engineering courses at two ESSLLI summer schools, and numerous individually guided student projects.

The LKB is an open-source grammar development environment implemented in Common-Lisp, and includes a parser, a generator, support for large-scale type hierarchies including the use of defaults, a rich set of graphical tools which both students and researchers can use for analyzing and debugging grammars, and full on-line documentation. Grammars of all sizes have been written using the LKB for several languages, mostly within the linguistic frameworks of Categorial Grammar and Head-Driven Phrase Structure Grammar. The system is distributed in both source code and as stand-alone binaries for several platforms which do not require a Lisp license.

The LinGO English Resource Grammar is an open-source broad-coverage bidirectional grammar of English, consisting of several thousand lexical and syntactic types organized hierarchically and a hand-built lexicon of 7000 stems. The grammar was originally implemented on the PAGE platform developed at DFKI Saarbrücken, and now runs on at least five additional platforms, including the LKB, where it is used between the groups as a representative reference for an emerging stan-

dard in typed feature structure formalisms, thus providing a common benchmark for ongoing collaborative work on efficient processing with such grammars. Much attention has been given in the grammar to the syntax-semantics interface, with precise meaning representations produced in the Minimal Recursion Semantics formalism.

Adapting the profiling metaphor known from software development to constraint-based language processing, the [incr tsdb()] profiling environment makes empirical assessment and systematic progress evaluation and comparison a focal point in the development cycle for grammars and processing systems. Based on (i) a set of structured reference data (both test suites and corpora), (ii) a uniform data model for test data and processing results, and (iii) a specialized profiling tool, developers are enabled to obtain an accurate snapshot of current system behaviour (a profile) with minimal effort. Profiles can then be analysed and visualized at variable granularity, reflecting various aspects of system competence and performance, and compared to earlier results. Since the [incr tsdb()] package has been integrated with some eight processing platforms (including the LKB and PET, see below) by now, it has facilitated cross-platform comparison and cross-fertilization between various research groups and implementations.

PET is a platform for experimentation with processing techniques and the implementation of efficient processors for unification-based grammars. It synthesizes a broad range of techniques for efficient processing from earlier systems in a modular ANSI C++ implementation. PET assumes the same descriptive formalism as the LKB and other systems processing the LinGO grammar. A parser built from PETcomponents can be used as a time- and memory-efficient run-time system for grammars developed in the LKB; in daily development of the LinGO and DFKI Japanese grammars it allows grammar engineers to frequently run rapid regression tests. Like the LKB, PET is integrated with [incr tsdb()] to support systematic experimentation, evaluation, and comparison.

We emphasize in this demonstration the crucial importance of multi-site collaboration to continue development of robust, reusable software for computational linguists to use in research and teaching. The components of the software suite we will demonstrate provide support for the benefits of such sustained collaboration. They are independent yet integrated, freely available, documented, and in use in multiple systems, at many sites, for multiple languages. All three systems support several Unix variants (including Linux and Solaris) and Microsoft Windows; additionally, the LKB is available for Mac OS. See 'http://lingo.stanford.edu/' for further details.

The demonstration is related to a pre-conference tutorial on practical applications of unification-based approaches held by Flickinger and Oepen, and to the expected presentation of a commercial product incorporating HPSG processing given by YY Software Corporation as part of the ACL Exhibit Programme.

Acknowledgements

Development of the LKB system was originally supported by ACQUILEX projects BRA-3030 and 7315 under the Esprit program (grant to Cambridge University). More recent research has been supported by the National Science Foundation under grant number IRI-9612682 (grant to Stanford University). Development of the LinGO ERG was also supported by NSF IRI-9612682 and by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the *Verbmobil* project under grant FKZ:01iV401. The [incr tsdb()] environment was developed as part of the Collaborative Research Division *Resource-Adaptive Cognitive Processes,* project B4 (PERFORM) funded by the *Deutsche Forschungsgemeinschaft* (grant to Saarland University); PET development was partly supported by a *Verbmobil* grant to the DFKI Language Technology Laboratory and also within the PERFORM project at Saarland University.