Towards Systematic Testing and Diagnosis Integrating TSNLP and ALEP

Stephan Oepen
Saarbrücken University
Postfach 151140
66041 Saarbrücken (Germany)
+49 - 681 - 302 41 76
oe@coli.uni-sb.de

Marius Groenendijk
Anite Systems
151, rue des Muguets
2167 Luxemburg
+ 352 - 43 95 94
marius@anite-systems.lu

Abstract

A recent addition to the ALEP grammar engineering platform is described: the test suite apparatus and test data produced in the TSNLP project have been seamlessly integrated with the ALEP task executor. The resulting test suite extension to ALEP is well-suited to substitute for the existing naive testing environment, greatly increases testing and report generation flexibility and fixes several (previously unknown) errors in the timing and coverage measures computed by the test suite processor. For downward compatibility the previous testing functionality is preserved in the current ALEP version (3.2 as of JAN-97).

1 Background: The ALEP, LSGRAM, and TSNLP Projects

There is only limited support for systematic testing and diagnosis in ALEP; currently only linguistic analysis is supported for testing. The testing processor is the so-called Analysis Test Suite Virtual Machine (anasuite vm). A test suite consists of a user-supplied input file with a number of test cases (as Prolog clauses) and a description of the grammars and lexicons (i.e. the linguistic environment) required to perform the linguistic processing on these sentences . The test suite processor produces a fairly detailed execution report including the processing and failure times for each of the linguistic processing steps. The test suite processor is batch-oriented: it can be launched from within ALEP but will continue to execute without the complete ALEP environment.

The LSGRAM project (and other projects) have made extensive use of this testing facility to do performance and coverage testing. The LSGRAM project has reached an impressive degree of coverage in terms of number of words but also in terms of various linguistic phenomena; the number of possible test cases is therefore potentially large. Hence, a generalized, more systematic and fully automated approach to testing in ALEP is required.

Recently, the TSNLP[1] (Test Suites for Natural Language Processing) project has delivered a multi-purpose test suite package addressing the increasing demand in the NLP community (among developers and users alike) for reference data to evaluate a rapidly growing number of natural language applications. The project produced the methodology and technology for the development and application of systematic test suites (see ?) and ?)). Substantial test suites of some 4500 items each were built for English, French, and German covering central syntactic phenomena. Test items are annotated using a rich annotation schema neutral to linguistic theories and specific types of applications. The TSNLP test suites are mounted on a relational database to maximally facilitate the maintenance, application, and adaptation of the data and to provide interfaces to both human test suite users as well as to arbitrary application programs.

2 Test Suite Usage in Grammar Engineering

Contemporary HPSG-type grammars with realistic grammatical and lexical coverage (e.g. the resources developed in LSGRAM) tend to have immense conceptual and computational complexity; as the grammatical framework aims to eliminate redundancy and factor out generalizations where possible, the interaction of lexicon and rule apparaturs can be rather subtle and make it difficult to predict how even modest changes to the grammar affect coverage and performace. Additionally,

[1] The project was started in December 1993 and completed in March 1996. Most of the project results (documents, bibliography, test data, and software) as well as on-line access to the test suite database and email addresses of the project members can be obtained through the world-wide web from the TSNLP home page 'http://tsnlp.dfki.uni-sb.de/tsnlp/'.

The TSNLP project was funded within the Linguistic Research Engineering (LRE) programme of the European Commission (DG XIII) under research grant LRE-62-089 and by the Swiss Federal Government.

in a distributed grammar engineering setup (i.e. for a project where several people or even sites contribute to a single grammatical resource) it is obviously necessary to assess the impact of individual contributions, frequently evaluate the quality of the overall grammar, and compare it to previous versions.

Test suites (i.e. systematic collections of artificially constructed and manually annotated reference data) have long been acknowledged as a suitable tool for fine-grained and focussed diagnosis, evaluation, and benchmarking (see e.g. ?) and ?)). Yet, traditional test suites typically were represented as flat files listing pairs of test sentences and grammaticality judgements plus, in some cases, informal section headings grouping sets of sentences according to linguistic phenomena (or sometimes application-specific criteria). The previously existing ALEP test suite apparatus is a prototypical instance of this type of unstructured test suites: although the choice of input file format (as ternary Prolog predicates) allows the automation of test suite processing and report generation, there is no support for processing only a subset of a test suite (e.g. only grammatical test items or those exemplifying a specific linguistic phenomenon), nor can the processor or report generator take advantage of the additional information (a comment string) associated with each item.

In comparison to first generation (or traditional) test suites, the TSNLP approach is novel in that the concept of a test suite as a monolithic set of test items has been abandoned in favour of a notion of a complex and highly structured test suite; the TSNLP test suites are implemented as a database in which test items are stored together with a rich inventory of associated linguistic and non-linguistic annotations. Thus, the test suite database serves as a virtual (or meta) test suite that provides the means to extract the relevant subset of the test data suitable for some specific task. Using the explicit structure of the data and the TSNLP annotations, the database engine allows searching and retrieving data from the virtual test suite, thereby creating a concrete test suite instance according to arbitrary linguistic and extra-linguistic constraints.

Moreover, the database concept allows to store application-specific coverage and performance measures as additional (distinct) relations (into the so-called *user & application profile*) that link the test data to a precise and contiguous record of how various system and grammar versions (at various times) perform; thus, the database concept easily facilitates progress evaluation, regression testing, and comparative report generation. At any point in the development cycle, grammar engineers can determine how changes to the grammar (e.g. adding a rule) or control information (e.g. in the ALEP case the choice of head declarations or the analysis – refinement partitioning) affect coverage and overgeneration or overall analysis performance; since test suite processing and report generation are fully automated, it can be done in an overnight batch job.

3 Integrating ALEP and TSNLP

To make the TSNLP diagnostic resources available within ALEP, in a cooperation between Cray Systems (now Anite Systems) and DFKI Saarbrücken the TSNLP database server tsdb has been seamlessly integrated with the ALEP kernel to provide the following functionality:

- browsing the set of available test suite databases;
- retrieval of test data from a test suite database allowing to select from the available test suites and their data on the basis of linguistic or extra-linguistic information captured in the TSNLP annotation schema;
- processing of TSNLP test data in batch mode;
- identification of precise and suitable coverage and performace measures and storage into a so-called user & application profile (i.e. a part of the test suite database itself);
- report generation based on current and previous performance measures (progress evaluation and regression testing); and
- deletion of outdated or otherwise unwanted data from the test suite database and especially the user & application profile.

The integration of TSNLP in ALEP has been performed according to the so-called tight-integration model. Integration has been performed on all possible integration levels in ALEP (?)): changes have been made (i) in the underlying linguistic processing machinery for the analysis test suite processor, (ii) on the ALEP User Language Level (AUL), and (iii) on the application level (scripts and executables). Furthermore, the TSNLP database front-end (tsdb) itself has been adapted to allow for seamless integration into the ALEP suite of applications. In the following sections we will describe how the integration was achieved in some detail (see ?) for further information).

3.1 Functional Interface

The basic principle behind the ALEP – TSNLP integration is quite simple. Access to TSNLP functionality of ALEP relies on the one hand on the presence of a tsdb server and on the other hand on the availability of a collection of shell scripts to provide a certain number of services. The collection of scripts is what we call the TSNLP back-end. The TSNLP back-end provides the following services.

- Start and stop the tsdb server;
- query the tsdb server in various ways (predefined queries, specific output et al.);
- update the data in the tsdb database; the results of a TSNLP-based analysis test suite run will record detailed coverage and timing data;
- produce various reports, most importantly the report generator for the test runs; his report generator produces a more detailed execution report than the original analysis test suite processor.

A number of new ALEP object types have been defined to support and access the new functionality. These TSNLP-specific ALEP objects are completed with the appropriate AUL tasks which operate on these objects.

To access the TSNLP databases, ALEP can communicate with the tsdb server through a socket-based interface. Once the connection between ALEP and the tsdb server is established, ALEP can send tsdb commands to the server and retrieve the results of these commands through a communication file. For instance, to start an analysis test suite, the user has the possibility to select or enter a tsdb query that produces a result (a number of sentences) that is suitable for input to the test processor. Once the test processor has been launched by the user, it will collect data about the linguistic processing. This data (time stamps, number of results et al.) is then send back to the tsdb back-end which updates the tsdb (viz. the ALEP user & application profile section).

3.2 User Interface

In ALEP it is possible to define new graphical user interface tools dedicated to a special task. This is achieved by configuring the 'Generic Tool' through AUL objects and tasks. A new GUI 'Test Suite Tool' has been defined which provides convenient access to most of the TSNLP functionality through a simple menu (the Actions menu). The TSNLP objects which can be used in that tool provide the access to the TSNLP functionality as sketched above.

Special emphasis is put on the interactive browsing and selection of subsets of test data from a tsdb database: before feeding a set of data to the test suite processor, the Test Suite Tool allows to input both predefined, partially free, and fully unrestricted tsdb queries to the database and view the query result(s) in an ALEP emacs(1) buffer; at any point, the current selection of test data can be converted into the input format for the test suite processor (i.e. the familiar Prolog `tx()` predicates) without recomputation.

4 Conclusion: Achieved Improvements and Outstanding Problems

The integration of the TSNLP package into ALEP not only provided the grammar engineering system with well-established, multi-lingual and multi-purpose test data of wide coverage and a systematic and at the same time flexible test suite apparatus, but also resulted in significant improvements to existing functionality as well as substantial new functionality.

Due to the work performed for the integration of the TSNLP database, a number of bugs in the traditional testing environment were found and fixed in the new approach. As the traditional report generator would often (and sometimes systematically) give inaccurate or misleading information, especially the computation of linguistic processing times is now more accurate and more fine-grained than before (note that the computation is now done by the TSNLP back-end rather than the test suite processor). The test reports produced jointly by the test suite processor and TSNLP back-end produce an equivalent report to the one produced by the originial processor. However, the data gathered during the processing is also stored in the TSNLP database. This stored information allows for the production of a more detailed processing report than before (since more data is recorded than actually exploited). Using appropriate tsdb queries, it is possible to produce new types of reports without having to code anything in the underlying ALEP test suite processor.

The integration of TSNLP completes some essential functionality in ALEP. For downward compatibility the original behaviour and functionality of the traditional test processor is intact and can be used exactly as before; in the long run, it is expected that ALEP users will want to import existing ALEP test suite files into the TSNLP database, add essential annotations (e.g. links to the TSNLP phenomena classification), and take advantage of the improved testing functionality.

5 Outlook: Future Work

Since the TSNLP database has already been integrated with other grammar engineering platforms (e.g. the DFKI PAGE environment used in, among others, the VerbMobil project), for the first time systematic and platform-independent diagnosis, testing, and benchmarking (allowing the comparison of results) becomes available. In the near future it is expected to verify the validity of the approach (and presumably identify its limits) in a controlled experiment with a few experienced ALEP and PAGE users. Deploying a newly available grammar compiler converting ALEP grammatical resources into the PAGE format, it should even be possible to perform a cross-platform evaluation between ALEP and the DFKI PAGE system. Based on a generalized notion of performance profiling for lexicalized constraint-based grammars (i.e. gathering very fine-grained performance and resource consumptions measures and do detailed statistical analyses), it is expected that the close coupling of contemporary grammar development platforms and the TSNLP database approach may allow for an improved understanding of the inherent computational complexity in processing HPSG-type grammars.

Given that TSNLP supports a very rich annotation schema to describe the test cases, it should be valuable to support this schema directly in ALEP too. Currently, very little of the TSNLP linguistic annotations are actually used in the test suite processing and report generation proper (though they may well have been the basis for selecting a specific subset of the data), nor is any of the processing result(s) (e.g. the phrase structure tree or a more abstract semantic representation) written to the TSNLP database for later reference or automatic comparison to a specification of the expected output. As a matter of fact, the actual result of processing a test item is immediately discarded by the processor. For a future version of the ALEP – TSNLP integration one can easily envisage to either feed the processing result to the TSNLP back-end for format conversion and extraction of relevant information, or use a tailored set of refinement (or transfer) rules for the purpose.

Besides, it should be investigated whether extension to the other linguistic operations could be supported by TSNLP (transfer and synthesis).

References

Flickinger, Daniel, John Nerbonne, Ivan A. Sag, and Thomas Wassow. 1987. Toward Evaluation of NLP Systems. Technical report. Hewlett-Packard Laboratories. Distributed at the 24th Annual Meeting of the Association for Computational Linguistics (ACL).

Groenendijk, Marius. 1994. *Configuration Guide. Customizing the* ALEP *Environment. Advanced* ALEP *User Guide*. Cray Systems and Commission of the European Community, 2.1 edition, August.

Lehmann, Sabine, Stephan Oepen, Sylvie Regnier-Prost, Klaus Netter, Veronika Lux, Judith Klein, Kirsten Falkedal, Frederik Fouvry, Dominique Estival, Eva Dauphin, Hervé Compagnion, Judith Baur, Lorna Balkan, and Doug Arnold. 1996. TSNLP — Test Suites for Natural Language Processing. In *Proceedings of COLING 1996*. Kopenhagen.

Nerbonne, John, Klaus Netter, Kader Diagne, Ludwig Dickmann, and Judith Klein. 1993. A Diagnostic Tool for German Syntax. *Machine Translation* 8:85–107.

Oepen, Stephan. 1996. ALEP – TSDB *Interface Specification*. Cray Systems and Commission of the European Community, August.

Oepen, Stephan, Klaus Netter, and Judith Klein. 1996. TSNLP — Test Suites for Natural Language Processing. In *Linguistic Databases*, ed. John Nerbonne. CSLI Lecture Notes. Center for the Study of Language and Information. forthcoming.