Sanskrit Linguistics Web Services

Gérard Huet
Inria Paris-Rocquencourt
gerard.huet@inria.fr

Amba Kulkarni
University of Hyderabad
apksh@uohyd.ernet.in

Abstract

We propose to demonstrate a collection of tools for Sanskrit Computational Linguistics developed by cooperating teams in the general setting of Web services. These services offer a systematic architecture integrating multilingual lexicons, morphological generation and analysis, segmentation and parsing, and interlink with the Sanskrit Library digital repository. They may be used as distributed Internet services, or installed as local tools on individual users workstations.

1 Community building

Sanskrit is the primary culture-bearing language of India, with a continuous production of literature in all fields of human endeavour over the course of four millennia. It benefited from a strong linguistics tradition, established from early times, and notably from the grammar composed by Pāṇini around the fourth century B.C.E., and commented since by innumerable grammatical treatises. This fairly complete descriptive apparatus took a prescriptive character, resulting in a constrained evolution of the language within its official grammar, leading to its stability as a semi-formal language. On the other hand, multiple styles of writing treatises, commentaries, and even poetry, led to a variety of specific dialects, both in prose and in versified form.

The efforts towards developing tools for the computational treatment of Sanskrit have been steadily progressing both at national as well as international level. A Sanskrit Computational Linguistics consortium funded by the Indian Government coordinates the development of consistent tools within 7 research institutes. In 2007, the first of a series of International Sanskrit Computational Linguistics Symposia was organized in Paris with the aim of gathering a community of teams sharing ideas as well as linguistic resources, and developing inter-operable software. These symposia have benefited the computer scientists from the grammatical expertise of the traditional scholars, while the traditional scholars could see the practical applications of the thousand of years old theories.

Within this general effort, specific tools were developed at Inria in Paris and University of Hyderabad for the analysis of Sanskrit texts, designed as inter-communicating Web services. A specific human-machine interface was developed, allowing annotation experts to produce tagged tree banks for the Sanskrit Library, a digital TEI-conformant repository of Sanskrit corpus. This joint work was presented at COLING-2012 (Goyal et al., 2012). We herein propose to demonstrate the current functionalities of this software platform.

2 Architecture of components

It was deemed counter-productive to attempt to build a monolithical rigid system, and we turned rather to developing on various sites independent components, communicating with each

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations, pages 48–51, Dublin, Ireland, August 23-29 2014.

other as Web services interchanging XML data. This allows freedom of programming languages and environments, operating systems, and even linguistic resources. This also permits greater flexibility of independent versioning of the components. Furthermore, HTML and client-side scripting provide a standardized solution to a common user interface, with easy multilingual display through Unicode.

Currently the system supports several lexicon resources. A specific core lexicon, the Sanskrit Heritage dictionary (Huet, 2004), has been developed as the seed resource for morphological databases. Digitalized versions of the Monier-Williams (Monier-Williams et al., 1999) and the Apte dictionaries are being progressively integrated by semi-automatic alignment. Amarakośa, the oldest thesaurus of Sanskrit, has also been digitalized (Nair and Kulkarni, 2010), providing the ontological information following the Indian tradition.

A number of components use the Zen Computational Linguistics Toolkit[1], a systematic functional programming library of finite-state tools. It handles lexicon management, phonological computation, morphology generation (both generative and inflexional), and segmentation. The segmentation component is specially important, in view of sandhi. Sanskrit text is represented as the result of phonetic smoothing, whose efficient inversion is problematic. An original solution to sandhi analysis was developed (Huet, 2005; Huet, 2006).

Morphological databanks are produced mechanically from the core dictionary stems, informed with their production parameters. Sharing techniques give highly compressed data structures loadable dynamically in process memory, eschewing the use of costly database technology. The morphological treatment processes are not strictly speaking Pāṇinian, but it is possible to relate them precisely to Pāṇinian derivations (Goyal and Huet, 2012). Independently, databanks issued from the Sanskrit traditional repositories have been digitalized (Bharati et al., 2006) and linked to the Aṣṭādhyāyī simulator which generates the nominal forms following the Pāṇinian process (Goyal et al., 2009). Lemma alignment algorithms permit inter-operability of these various resources, seen as plug-in components.

Segmentation leads to morphological tagging, a complete but much over-generating process. A typical sentence may have billions of possible analyses. A graphical user-interface, providing a fully shared view of all segmentations, has been developed (Huet and Goyal, 2013). It is designed to be very fast, and to allow human annotators easy inspection of the segment features. Experiments with semi-automatic annotation of parts of the Sanskrit Library[2] validated the approach.

A deterministic tabulated dependency parser has been designed (Kulkarni, 2013). The application of local constraints at an early stage and stacking of intermediate results along dynamic programming yield fast results. The graphical user-interface of the segmenter is adapted to the parser in order to show the shared view of all possible solutions in a compact form.

Annotated corpus statistics are being used to build the language and grammar models for various tasks. An experimental version of a segmenter (Kumar et al., 2010) is developed that constraints the sequence of constituents by a language model and uses the split model based on the empirical data of sandhi rules for splitting. These empirical models may further be used to build weighted finite state automaton to prioritize the solutions.

3 Salient novel features

The segmentation algorithm uses a novel approach to finite state technology, through Effective Eilenberg machines (Huet and Razet, 2008). Although Sanskrit has huge literature, only a negligible part of it has been tagged for various levels of analyses. Thus use of statistical techniques or machine learning is almost ruled out. While machines are good at syntactic analysis, for semantic compatibility of solutions we still depend on human assistance. This calls for a suitable interface which can represent billions of solutions with all relevant linguistic details

[1] `http://yquem.inria.fr/~huet/ZEN/`
[2] `http://sanskritlibrary.org`

to be displayed on a screen. This requirement led us to develop a tabulated display interface, using efficiently a compact shared representation of solutions, presenting an ergonomical solution to human assistance. This interface was also further adapted to display all possible sentential parses in a compact tabular format for choosing the correct parse.

A new technology of forms alignment, indexed by their morphological production history ("unique naming"), allows uniform access to various dictionaries, despite possibly conflicting homophony partitions.

The consistently structured core lexical repository, together with lexicon alignment, allows the automatic production of derived human-readable dictionaries under the Babyloo/Stardict/Goldendict formats, with consistent hypertext linking to grammatical processes.

Some requirements of Sanskrit computational tools are very specific. Sanskrit has a vast literature spreading over several knowledge domains. Most of the important Sanskrit literature is already translated into several languages. In spite of this, scholars want to have access to original sources, and thus development of the computational tools with convenient user interfaces that allow seamless connectivity to and from the lexical resources, generation engines and analysis tools becomes meaningful. Further, the availability of Aṣṭādhyāyī, an almost complete grammar for Sanskrit, also puts demands on the developers to authenticate the inverse process of analysis by the generative rules of grammars. These considerations have resulted in the development of suitable interfaces linking various resources and tools through Web services.

4 Software engineering and deployment issues

The Web services approach allows independent development of components, seen as XML transducers keeping a history of interactions through the argument structure of the CGI invocations. This allows independent development, archiving and distribution of modules developed in C, PERL, Ocaml, Python, Java, Javascript, etc.

Parametrization of the various platform interconnections allows for distributed use through Internet, as well as local use on workstations. Extreme programming methodology allows for agile development with high frequency releasing and a fast user feedback.

The software, as well as linguistic resources, are available under open-source licences.

5 Demonstration scenario

The tools will be demonstrated on a few typical sentences, showing various usages of the software. The first presentation will demonstrate the Sanskrit Heritage segmenter on an input sentence. It will show how to select a segmentation solution using the graphical interface, then the way to refine the solution using the dependency parser of the Hyderabad Sanskrit Computational Linguistics analyser, in order to get its dependency structure. A dual presentation will start from the Hyderabad analyser, using the Heritage segmenter as a front end. Finally it will be shown how to access the analyser tools from marked-up corpus in the Sanskrit library. Settings allow switching between the various lexicons, and displaying grammatical information either in romanized Western style, or in *Devanāgarī* traditional Indian style. The demo will also include linking to actual Pāṇinian derivation process to ensure precision in the analysis. If time permits, the Goldendict versions of the lexicons will be shown, informed with grammatical information.

Acknowledgement

The Inria 'Sanskrit Heritage' platform benefited from important contributions of Pawan Goyal, notably in its graphical interface. We wish also to thank Peter Scharf for his cooperation on the Sanskrit Library interface. Various components of the software at University of Hyderabad were developed with support from TDIL Programme, DeitY, Government of India for the project 'Development of Sanskrit computational toolkit and Sanskrit-Hindi Machine Translation system' with contributions from Sivaja Nair, Anil Kumar, Karunakar, Devanand Shukl and Pavankumar.

References

Akshar Bharati, Amba Kulkarni, and V. Sheeba. 2006. Building a wide coverage Sanskrit morphological analyser: A practical approach. First National Symposium on Modeling and Shallow Parsing of Indian Languages, IIT Mumbai.

Pawan Goyal and Gérard Huet. 2012. Completeness analysis of a Sanskrit reader. In *Proceedings, 5th International Symposium on Sanskrit Computational Linguistics*. DK Publisher.

Pawan Goyal, Amba Kulkarni, and Laxmidhar Behera. 2009. Computer simulation of *Aṣṭādhyāyī*: Some insights. In Gérard Huet, Amba Kulkarni, and Peter Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 139–161. Springer-Verlag LNAI 5402.

Pawan Goyal, Gérard Huet, Amba Kulkarni, Peter Scharf, and Ralph Bunker. 2012. A distributed platform for Sanskrit processing. In *24th International Conference on Computational Linguistics (COLING), Mumbai*.

Gérard Huet and Pawan Goyal. 2013. Design of a lean interface for Sanskrit corpus annotation. In *Proceedings, ICON13, Hyderabad*.

Gérard Huet and Benoît Razet. 2008. Computing with relational machines. ICON'2008 tutorial.

Gérard Huet. 2003. Towards computational processing of Sanskrit. In *International Conference on Natural Language Processing (ICON)*.

Gérard Huet. 2004. Design of a lexical database for Sanskrit. In *Workshop on Enhancing and Using Electronic Dictionaries, COLING 2004*. International Conference on Computational Linguistics.

Gérard Huet. 2005. A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *J. Functional Programming*, 15,4:573–614.

Gérard Huet, 2006. *Themes and Tasks in Old and Middle Indo-Aryan Linguistics, Eds. Bertil Tikkanen and Heinrich Hettrich*, chapter Lexicon-directed Segmentation and Tagging of Sanskrit, pages 307–325. Motilal Banarsidass, Delhi.

Gérard Huet. 2007. Shallow syntax analysis in Sanskrit guided by semantic nets constraints. In *Proceedings of the 2006 International Workshop on Research Issues in Digital Libraries*, New York, NY, USA. ACM.

Amba Kulkarni and Devanand Shukl. 2009. Sanskrit morphological analyser: Some issues. *Indian Linguistics*, 70(1-4):169–177.

Amba Kulkarni, Sheetal Pokar, and Devanand Shukl. 2010. Designing a constraint based parser for Sanskrit. In G N Jha, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465.

Amba Kulkarni. 2013. A deterministic dependency parser with dynamic programming for Sanskrit. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 157–166, Prague, August. Charles University Matfyzpress, Prague, Czech Republic.

Anil Kumar, Vipul Mittal, and Amba Kulkarni. 2010. Sanskrit compound processor. In G N Jha, editor, *Proceedings of the International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465.

Vipul Mittal. 2010. Automatic sanskrit segmentizer using finite state transducers. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 85–90, Uppsala, Sweden, July. Association for Computational Linguistics.

M. Monier-Williams, E. Leumann, and C. Cappeller. 1999. *A Sanskrit-English Dictionary: Etymological And Philologically Arranged With Special Reference To Cognate Indo-European Languages*. Asian Educational Services.

Sivaja S. Nair and Amba Kulkarni. 2010. The knowledge structure in Amarakośa. In G N Jha, editor, *Proceedings of the International Sanskrit Computational Linguistics Symposium*. Springer-Verlag LNAI 6465.

Peter Scharf and Malcolm Hyman. 2009. *Linguistic Issues in Encoding Sanskrit*. Motilal Banarsidass, Delhi.

S.C. Vasu. 1980. *The Aṣṭādhyāyī of Pāṇini*. Motilal Banarsidass.