Design of an Electronic Sanskrit Reader

Extended abstract - SALA XXI

Gérard Huet

INRIA-Rocquencourt

Abstract

We present the design of a platform for computational linguistics applied to classical sanskrit. The platform is anchored on a structured lexical repository. A toolbox for morphology allows generation of flexed forms and sandhi analysis through finite automaton technology, leading to segmentation and lemmatisation. Computation of taggings consistent with concord and verb valencies constraints leads to a first level of shallow parsing, during which occurs compound analysis. This leads to a proposal for semi-automated corpus annotation. The talk will be illustrated by concrete examples on a running system.

1 Introduction

We sketch the design of a computer program that aims at automating the reading and understanding of a sanskrit text. Such a program may be useful for a scholar preparing a critical edition of some sanskrit corpus.

By reading and understanding we mean the following. We first assume the corpus of interest has been digitalised in the form of a text file representing some encoding of say a *devanāgarī* document. The precise encoding is of little importance, it may for instance be an ASCII sequence of letters representing one of the standard roman transliteration conventions such as the Kyoto-Harvard one, or the Velthuis one. It may also be a direct Unicode encoding of the *devanāgarī* text. All such representations are easily inter-convertible in linear time using an easy to define regular transducer. However, we stress that a document digitalised as a mere photographic scan of a manuscript will need to go through an initial processing, be it optical script recognition or human typing.

The first stage of processing is to return a list of tagged sentences. A tagged sentence is a sequence of word forms such that the original sentence

may be obtained from it through euphony transformation (i.e. *sandhi*). This corresponds to what the sanskrit grammarians call the *padavibhāga* analysis of the sentence. Each word form may then be itself analysed as a non-empty set of pairs ⟨ stem,attributes ⟩ , where stem is a dictionary entry, and attributes indicates whether the word is un undeclinable form, a declined substantive (adjective, pronoun, number) with its gender and number, or a finite verb form with its mode, tense, person and number. Let us give an example. The *devanāgarī* sentence *kusuma.mgopiibhya.hk.r.s.nodadati*, or equivalently *kusumaṃgopībhyaḥkṛṣṇodadati* in roman transliteration with diacritics, will result in the following *padavibhāga* analysis:
⟨*kusumam*⟩⟨*gopībhyas*⟩⟨*kṛṣṇas*⟩⟨*dadati*⟩ where each of the 4 word forms corresponds to a set of taggings as follows: *kusumam*={(*kusuma* n. nom. sg.), (*kusuma* n. acc. sg.), (*kusuma* n. voc. sg.)}
gopībhyas={(*gopī* f. abl. pl.), (*gopī* f. dat. pl.)}
kṛṣṇas={(*kṛṣṇa* m. nom. sg.)}
dadati={(*dā* ind. pr. 3 sg.)}.

The second stage of processing consists in grouping the words in simple phrases, each containing at most one finite verb form, and for each such phrase, choosing unique tags, in such a way that the corresponding roles are consistent with the verb grammatical valency. In our simple example the verb *dā* (to give) has a possible valency ⟨nom. acc. dat.⟩ , corresponding to the respective roles of subject, direct object and indirect object, and there is a unique way to choose taggings consistently with this valency, namely
kusumam=(*kusuma* n. acc. sg.)
gopībhyas=(*gopī* f. dat. pl.)
kṛṣṇas=(*kṛṣṇa* m. nom. sg.)
dadati=(*dā* ind. pr. 3 sg.)
yielding the expected meaning (Kkṛṇa gives a flower to the maidens).

Note that the sandhi transformations are successively: $m|g \rightarrow$*ṃg,* $s|k \rightarrow$*ḥk*, $as|d \rightarrow od$.

It should be stressed that several possible *sandhi* decompositions may be possible in general, leading to several solutions of the first phase. This is due to the non-deterministic nature of *sandhi*, and is actually unavoidable, since sanskrit poets took advantage of this inherent ambiguity in order to compose works with double entendre. For each such solution, many choices may remain after the second phase. For instance, a verb accepting two accusative complements will usually result with two readings, whose disambiguation may require semantic informations, such as the distinction between animate and inanimate objects. Finally, homonyms (i.e. homophons) may also lead to ambiguities, although more rarely.

Even for a simple sanskrit sentence in the indicative mode, tagging the words with attributes such as their case is not enough to determine their grammatical role. Accusatives, instrumentals and ablatives are often used as adverbs. Genitives may be either genuine genitive complements in a nominal phrase or indirect objects of certain verbs. Finally compound recognition is complex, since a compound stem may be confused typically with a vocative. Furthermore, compounds arise in several roles, such as copulative compounds (*dvandva*), *descriptive* compounds, where the left element is an attribute qualifying the right one (*karmadhāraya*), and *determinative* compounds, which act as (exocentric) adjectives (*bahuvrīhi*).

The end result of the second phase ought really to be a *dependency structure*, which then constitutes a reasonably complete syntactic analysis, since word order is not really important for the sense, and may rather be chosen for the prosody.

It is expected that the computation of dependencies will be trained and tuned from corpus analysis. Our envisioned computational platform is thus composed of three main components. The first one is a lexicographic data base. The second one is a corpus data base. The third one is a text analysis interpreter. Various other modules act as service components: finite state transducers of various kinds, morphology processors, constraint satisfaction engines, proof search tools, etc. Details of control flow between the various components will depend according to the task aimed at: text annotation, discourse management, translation, etc.

2 The current tools

We briefly sketch the current status of our sanskrit platform.

2.1 The lexicon

The lexical data base is the central component. The initial version of the data base was created by reverse engineering from a text lexicon developed over the years by the author. At the time of writing, it comprises 13000 entries. This lexicon compiles into two very different representations:

- a high-quality printable 320 pages sanskrit-to-french dictionary, with entries listed both in *devanāgarī* and romanisation with diacritics
- a web site giving an hypertext view of the same document.

Here is a sample entry in the first format (in one column format, the actual dictionary uses a two-column format):

> कर्मन् *karman* [*kṛ*$_1$-*man*] m. n. [sg. nom. *karmā*; acc. *karmānam*; i. *karmanā*; en comp. *karma*] acte, action, œuvre; fait, exécution, opération; cérémonie, sacrifice | ouvrage; occupation, métier; destin | fonction individuelle, devoir de sa caste | phil. accumulation de mérites et de fautes au cours des existences passées; destinée | Acte suprême, Œuvre sainte | pratique (opp. théorie); syn. *prayoga* | gram. le rôle [*kāraka*] objet (accusatif ou nominatif en mode passif) |phil. [*vaiśeṣika*] la catégorie [*padārtha*] des activités; la tradition en donne 5: *utkṣepaṇa* l'élévation, *avakṣepaṇa* l'abaissement, *ākuñcana* la contraction, *prasāraṇa* l'expansion et *gamana* le déplacement.

2.2 The euphony processor

An internal sandhi algorithm has been written, which allows morphological derivations from stems, prefixes and suffixes. Such transformations, involving complex retroflexion analysis, may be complex, allowing for instance to derive *līḍha* from *lih+tha* or *buddha* from *budh+ta*.

2.3 The grammatical engine

The declension tables for substantives/adjectives, pronouns, and numbers have been entered, leading to 84 tables of 8 cases times 3 numbers, with numerous exceptions. The conjugation tables of verbs are yet to be written, though, to complete the morphology production.

2.4 The flexed forms lexicon

A lexicon of all flexed forms may be generated, by piping the grammatical engine into the lexicon stems. For the moment, it comprises about 100000 forms. A reverse table gives for each of the forms its possible morphological origins. This information fits in 3MB of memory, and may thus be manipulated directly as a data structure by current days PCs.

2.5 Indexes to the lexicon

The above tools are combined in order to derive various search tools of the web site, notably an index engine returning all entries (and the declensions thereof) matching a given prefix, a simplified index where diacritics may be omitted from the query, and a grammatical engine computing declensions of a given stem with a given gender.

The grammatical engine is itself directly accessible from the Web lexicon, by clicking on the gender declarations.

2.6 The segmenting transducer/tagger

A generic tool has been designed for analysing euphony rewrites rules, and applied to generate a finite-state machine from the flexed forms lexicon in order to analyse external sandhi. For each entry sentence, the analyser generates the various ways in which flexed forms from the lexicon may combine with external sandhi in order to produce the input. Thus, on input *oṃnamaḥśivāya* it finds a unique solution :
[*om* with sandhi $m|n \rightarrow$ *ṃn*]
[*namas* with sandhi $s|$*ś*$\rightarrow$*ḥś*]
[*śivāya* with no sandhi].

In a tagging mode, the segmenter gives the various tagging possibilities. With the example above, we get:

```
Chunk: o.mnama.h"sivaaya
may be segmented as:

 Solution 1 :
[  om
 < und. form of om
 >  with sandhi m|n -> .mn]
[  namas
 < iic. form of namas#1
 | acc. sg. n. of namas#1
 | nom. sg. n. of namas#1
 | voc. sg. n. of namas#1
 | und. form of namas#1
 >  with sandhi s|"s -> .h"s]
[  "sivaaya
 < dat. sg. m. of "siva
 | dat. sg. n. of "siva
 >  with no sandhi]
```

To come back to the example at the beginning at this paper, replacing *dadati* (finite verb form, not yet entered in the flexed dictionary) by the participle *dadat*, we get as expected:

```
Chunk: kusuma.mgopiibhya.hk.r.s.nodadat
may be segmented as:

[  kusumam
 < acc. sg. n. of kusuma
 | nom. sg. n. of kusuma
 | voc. sg. n. of kusuma
 >  with sandhi m|g -> .mg]
[  gopiibhyas
 < abl. pl. f. of gopa
 | dat. pl. f. of gopa
 >  with sandhi s|k -> .hk]
[  k.r.s.nas
 < nom. sg. m. of k.r.s.na
 >  with sandhi as|d -> od]
[  dadat
 <...>  with no sandhi]
```

In certain cases there is overgeneration, and too many "solutions" are proposed to the user. For instance, *kāntārevanadurgeṣu* yields 39 solutions. In such cases, one may help the segmenter by providing blanks. For instance, *kāntāre vanadurgeṣu* yields only 6 segmentations for the first chunk, and 3 for the second.

2.7 Experimental release of the tools

All the tools described above are accessible on Internet, as various downloadable documents or executable CGI interfaces, at URL:
`http://pauillac.inria.fr/~huet/SKT`.

3 Future work

The next immediate task is to implement the morphology of verbs, in order to complete the dictionary of flexed forms.

The basic euphony analysis program needs to be adapted. Many particle and other tool words are overgenerative, and they must be removed as independent segment generative entities, either because they are redundant with morphology formation rules, or because they ought to be treated in a higher layer of syntax.

Compounds are treated piecewise, which is logical since the sandhi between compound components is basically the same as external sandhi between words. So we only need to enter in the flexed lexicon the stem forms which are used as leftmost component of a component, as a supplementary case so to speak. However, this method is incomplete for *bahuvrīhi* compounds which admit other genders than their rightmost component. For instance, *raktabīja* "regenerated by his blood" is a masculine mythological monster, yielding sg. nom. *raktabījaḥ*, while *bīja*, a neuter substantive, does not generate *bījaḥ*. This problem may be fixed in various manners, but it is not obvious at this point which is the best. It is clear in any case that a lot of tuning with actual corpuses will be necessary before obtaining a usable complete sanskrit analyser which does not not excessively overgenerate.

The next step will then be the building of a specially robust tagging mode, which will fail gracefully in the presence of words absent from the lexicon, in order to obtain a tool to semi-automatically complete the lexicon from a corpus.

References

[1] Abel Bergaigne. "Manuel pour étudier la langue sanscrite." F. Vieweg, Paris, 1884.

[2] Akshar Bharati, Vineet Chaitanya and Rajeev Sangal. Natural Language Processing - A Paninian Perspective. Prentice-Hall of India, New Delhi, 1995.

[3] M. B. Emeneau. "Sanskrit Sandhi and Exercises". University of California Press, 1964.

[4] Christopher J. Fynn. How to use Diacritics for Romanised Indic Text on the WWW. `http://www.user.dicon.co.uk/~cfynn/sanskrit.htm`.

[5] Gérard Huet. "Structure of a sanskrit dictionary." INRIA Technical Report, oct. 2000. Available as `http://pauillac.inria.fr/~huet/PUBLIC/Dicostruct.ps`.

[6] Gérard Huet. "Computational Linguistics for Sanskrit: a Software Engineering Approach." Submitted for publication, 2001.

[7] Gérard Huet. "Tagging a Stream of Phonems by Lexicon-directed Euphony Analysis – Application to a Sanskrit Lemmatizer." Submitted for publication, 2001.

[8] Gérard Huet. "From an informal textual lexicon to a well-structured lexical database: An experiment in data reverse engineering." IEEE Working Conference on Reverse Engineering (WCRE'2001), Stuttgart, Oct. 2001.

[9] Ronald M. Kaplan and Martin Kay. "Regular Models of Phonological Rule Systems." Computational Linguistics (20,3), 1994.

[10] Brett Kessler. "External sandhi in Classical Sanskrit." M.S. Thesis, Stanford University, Nov. 1992.

[11] Brett Kessler. "Sandhi and Syllables in Classical Sanskrit." Proceedings, Twefth West Coast Conference on Formal Linguistics, E. Duncan, D. Farkas and P. Spaelty, (eds.), p. 35–50. CSLI, Stanford, Calif.

[12] Anshuman Pandey. Devanāgarī for TEX, version 1.6, Aug. 1998. Available from any CTAN archive site, such as `ftp://ftp.tex.ac.uk/`.

[13] Xavier Leroy et al. "Objective Caml." See: `http://caml.inria.fr/ocaml/index.html`.

[14] William Dwight Whitney. "Roots, Verb-forms and Primary Derivatives of the Sanskrit Language." 1885. Repr. Motilal Banarsidass, Delhi, 1997.

[15] William Dwight Whitney. "Sanskrit Grammar." 5th edition, Leipzig, 1924.