Sanskrit Segmentation

Gérard Huet

INRIA Paris-Rocquencourt, France
Gerard.Huet@inria.fr

Paper presented at SALA XXVIII

1 Introduction: Sanskrit text presentation and preprocessing

We discuss in this paper the topic of Sanskrit segmentation, that is how to solve by computer software the problem of identifying in a Sanskrit sentence the division of a continuous enunciation into a sequence of discrete word forms. This is the first layer of Sanskrit computational linguistics.

First of all, we shall not deal here with speech recognition. We assume that our Sanskrit text is represented as a *devanāgarī* written input. We furthermore assume, in agreement with Classical Sanskrit corpus, that the accent is not indicated. This allows us to profit of the availability of this corpus to aim at building a tool of use for philologists to assist them in analysing such corpus; firstly in effecting the first stage of analysis, that of transforming the continuous enunciation (*vākyam*) into a stream of morphological items, the so-called *padapāṭha*; secondly, hopefully, in providing deeper levels of analysis of the Sanskrit utterance, some syntactic or semantic representation. Of course, the absence of accent and other prosodic markers is a handicap. Thus we shall have no way to distinguish in the compound *indraśatru* its *tatpuruṣa* interpretation *indraśatrúḥ* "Indra's slayer" from its *bahuvrīhi* one "*índraśatruḥ* "Slain by Indra" – just the opposite meaning ! This example shows by the way that we should not expect too much from a mechanical Sanskrit analyser. After all, we cannot hope that a computer will be smarter than the priests officiating at *Tvaṣṭā*'s *yajña*...

Digitalized Sanskrit corpus exists mostly in transliterated representation. A new trend in expressing *devanāgarī* in Unicode representation is emerging, but this does not seem an optimal solution from the information sciences point of view. Unicode is good for graphical representation, either on screens or on paper, using uniform rendering engines linked with modern formats of fonts. But Unicode encodings such as UTF-8 are notoriously bad at string algorithms, because of their variable width characters, slowing down needlessly the processing algorithms. Furthermore the Unicode encoding of *devanāgarī* mixes glyphs and phonemes together with control characters such as *virama*, a confusion between logical structure of syllables and glyph composition of ligatures. Thus we shall in the following assume input to be some ASCII character file representing transliterated text.

Of course there are many competing systems for Sanskrit transliteration. Some groups prefer transliteration schemes where each phoneme is represented one-to-one by a Latin character, leading sometimes to weird-looking encodings. Of course, this has the slight advantage that transliteration is thus trivially a prefix code, exempt from ambiguities. Several popular transliteration schemes, such as Velthuis' s scheme, which we shall use in our examples, write *ai* and *au* for the long diphtongs, leading possibly to ambiguities, albeit only for rare words with internal hiatus, such as *pra_ucya*. Anyway, the choice of one transliteration scheme or another is completely irrelevant to the problems of text processing such as segmentation. Files encoded in one scheme may be transduced in any other scheme by very fast linear time transducers easily available as Perl scripts or others, so this is a non-issue.

What matters, however, is a lot of detail about marking hiatus or giving segmentation hints such as the *avagraha*, noting elision of an initial *a* by sandhi. It is usual for helping the human reader to separate words by spaces, but then there are two completely distinct traditions to present the phonemic segments. One is so-called the *padapāṭha*, where sandhi has been undone. The other one is to break the stream of phonemes by blanks without undoing sandhi, respecting the *devanāgarī* reading, such as *yad iha asti tad anyatra yan neha asti na tat kvacit*, to be compared with its un-sandhied pre-*padapāṭha* form *yat iha asti tat anyatra yat neha asti na tat kvacit* "What is here (i. e. in the *Mahābhārata*) is found elsewhere, what is not here is nowhere to be found". We would like to profit of these segmentation hints (to the extent that we indeed trust them) and undo sandhi separately on each chunk rather than on the full *yadihāstitadanyatrayannehāstinatatkvacit*, which is unavailable anyway since unreadable (of course the original *devanāgarī* is readable, but human reading proceeds by direct identification of syllable groupings by their glyphs, and this is lost in transliteration).

The difficulty is that the problem is somehow ill-posed, like unfortunately most problems in real life. There is no systematic solution to the presentation of transliterated text with segmentation hints. For instance, hand segmentation may leave compounds whole, or may break them into constituents. For instance, the critical edition of the *Mahābhārata* as transliterated by the team of Pr Tokunaga indicates segmentation points inside compounds with dots, such as *śiva.liṅgam*. The Clay Sanskrit library uses its own specific notation. Actually, even the use of *avagraha* in *devanāgarī* text is itself a late convention, and there seems to be no generally agreed standard of its use.

One important fact to notice is that transliterated Sanskrit with blank space is essentially ambiguous, because some (most) spaces indicate just a segmentation hint in sandhied text, whereas some indicate actual hiatus in the *devanāgarī* text. For instance, consider the following 'chicken logic' aphorism from Kumārila (*kukkuṭanyāya*):[1]

na hi kukkuṭyā ekadeśaḥ pacyata ekadeśaḥ prasavāya kalpate[2]

[1] communicated by A. Aklujkar.

[2] we cannot assume the chicken to be half cooking and half laying eggs.

Here two of the blank spaces stand for hiatus, and their preceeding words in unsandhied form are *kukkuṭyāḥ* and *pacyate*. The other spaces are phonetically void. This example shows that it is not possible to undo sandhi independently in the transliteration strings separated with spaces. This problem is described at length in the appendix to [10], where a notion of *chunk* is proposed in a preprocessing phase, so that it is possible to undo sandhi independently on chunks. We shall not repeat here this discussion, which is a bit fastidious.

Another preprocessing problem concerns the *anusvāra* nasalisation sign. Besides the so-called genuine *anusvāra* in front of spirants and 'r', *anusvāra* in front of stops is more or less equivalent to the homophonic nasal of such. Thus we may encounter either spellings *sandhi* or *saṃdhi*, and if we do not want to duplicate all such forms in our lexicons we have to assume some standardisation, thus non genuine *anusvāra* is assumed to be replaced when possible by the homophonic nasal, like in *sandhi*. We remark in passing that it would be incorrect to write *sasaṃja* for *sasañja*, a perfect form from root *sañj*.

Still another problem is that of common degeminations by scribes, writing *tatva* for *tattva*, *satva* for *sattva*, *chātra* for *chāttra* and the like. Whether such forms are Pāṇinian or not seems open to debate. Sometimes one also encounters weird geminations, like *kīrttyate* for *kīrtyate*, this time sanctified by Pāṇini.[3] In any case, such variations are rather frequent, and although the problem is rarely evoked, it must be dealt with.

We shall not discuss further these preprocessing problems, which is an artifact of scribe conventions and philology traditions rather than the essential segmentation problem of continuous *devanāgarī* text, to which we now turn.

2 Analysis of external sandhi

Words in a sentence are joined by external (*bahiraṅga*) sandhi. Thus a first approximation of our segmentation problem is to find a sequence of Sanskrit words such that, after applying external sandhi at their junction, we obtain the input sentence. This decomposes into two concerns, firstly to construct a lexicon of Sanskrit (inflected) words complete in the sense that it contains the vocabulary of the target corpus, and secondly to design an algorithm able to decompose an input sentence into words from the lexicon glued together by (external) sandhi. This decomposition is not unique in general. For instance[4] the small sentence *śvetodhāvati* may be decomposed into the two words *śvetaḥ* (white) and *dhāvati* (runs) joined together with sandhi rule $aḥ \mid dh \rightarrow odh$, and meaning "the white (one, horse, etc.) runs". But *śvetaḥ* itself may be decomposed further into *śvā* (dog) and *ita* (here), joined together with sandhi rule $ā \mid i \rightarrow e$, yielding another meaning "the dog runs (towards) here". Thus we are stuck with non-determinism right at the start of linguistic analysis, and we cannot aim for a unique solution to our problem. We have to construct a set of potential solutions, and somehow use the context or whatever extra information in order to tell the dog from the

[3] as pointed out by P. Scharf, in *sūtra* 8.4.46, hopefully an optional rule

[4] this example, originally due to Patañjali, was communicated by J. Houben.

horse and choose the relevant meaning. And of course, beyond such genuine ambiguities, we may expect to obtain myriads of nonsense sequences of words, and indeed we do.

Fortunately, external sandhi may be analysed completely in a finite way. More precisely, under a mild assumption (the so called non-overlapping condition), there exists a finite number of solutions to the inverse of a junction transduction, a special case of transducers definable by finite-state automata under which falls external sandhi. This problem is analysed mathematically in [7], where an algorithm is detailed for sandhi analysis, implemented via a finite-state machine compiled from the lexicon. Further publications explain that the solver has good modularity properties [11], which may be exploited here in splitting the lexicon into lexical categories, allowing a finer grain analysis of words, the enforcement of a certain geometry of morphological construction, and variations in sandhi rules according to morphological transitions.

Thus we may now distinguish words between the three main lexical categories of Sanskrit, namely verbal forms, substantival forms, and indeclinables. Let us for instance show the top-down view of the control graph of a simplified version of our segmenter.

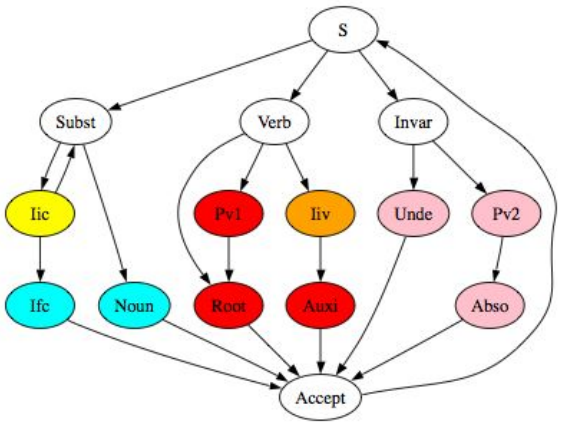

Fig. 1. The 10-phases lexical analyser.

We can read on this figure the finite-state description of a Sanskrit sentence S understood as a sequence of Words, where a Word is either a Substantive, a Verb, or an Invariable form. In turn a Substantive is an inflected Noun, possibly prefixed by an arbitrary list of compound-forming noun stems (Iic), or an inflected noun root form (Ifc), necessarily prefixed by a non-empty list of compound-forming noun stems. This last distinction is absolutely necessary in order to avoid bad over-generation of wrong segmentations, where the suffix-formations of *-pa*, *-ga*, *-da* and the like would be attempted as substrings of

all kinds of words if they were allowed as stand-alone. Due to their shortness, these root forms must be restricted to right-components of compounds (except for exceptional roots such as *bhū* and *yudh*). Next, a Verb may be a root form, possibly prefixed by a sequence of preverbs (Pv), or it may be a composite form of an auxiliary root finite form, with a special periphrastic form of a substantive formed with an *-ī* suffix (technically designated as *cvi* in Pāṇinian terminology). We coined for them Iiv (*in initio verbum*) by analogy with Iic standing for *in initio compositi*, a common Western categorization. Such compound forms are rarely described in grammars, although they form very common expressions such as *ghanī bhū* (to harden), *dūrī bhū* (to go away), *dūrī kṛ* (to send away), *navī kṛ* (to renovate), *pavitrī kṛ* (to purify), *bhasmī kṛ* (to reduce to cinders), *mithunī as* (to mate) etc. In order for such compound verbal forms to be recognized as legal Sanskrit, the lexical analyser must accommodate these *-cvi* morphemes. Rather than trying to have an exhaustive list of such idiomatic usage in our lexicon, and an associated complex sub-automaton, we decided to make this lexical category productive, and to duplicate the finite root forms of auxiliaries *bhū*, *as* and *kṛ* (node Auxi in Fig. 1). In a similar spirit of simplicity we decided not to limit preverb prefixes for a given root to the ones explicitly listed in the lexicon, although the corresponding data structure is indeed explicit in our implementation. Instead, we collect all preverb prefixes used for at least one root, and allow to recognize them as prefixes of any root. The slight overgeneration which this induces is compensated in our opinion, firstly in the simplified and thus more compact automaton, and secondly in the generative capacity with which it recognizes preverb formation. Our machine will recognize some verb forms as legal even when the corresponding entry does not exist in our by essence limited lexicon.

Maybe this list of attested preverb sequences is worth listing explicitly, since to our knowledge this information is not readily available in standard grammars:[5] *ati*, *adhi*, *adhyava*, *adhyā*, *anu*, *anuparā*, *anupra*, *anuvi*, *antar*, *anvā*, *apa*, *apā*, *abhi*, *abhini*, *abhipra*, *abhivi*, *abhivyā*, *abhisam*, *abhyanu*, *abhyava*, *abhyā*, *abhyut*, *abhyupa*, *abhyupā*, *ava*, *ā*, *āpra*, *ut*, *utpra*, *udā*, *upa*, *upani*, *upasam*, *upā*, *upādhi*, *tiras*, *ni*, *nis*, *nirava*, *nirā*, *parā*, *pari*, *parini*, *parisam*, *paryupa*, *puras*, *pra*, *prati*, *pratini*, *prativi*, *pratisam*, *pratyapa*, *pratyabhi*, *pratyava*, *pratyā*, *pratyut*, *prani*, *pravi*, *pravyā*, *prā*, *bahis*, *vi*, *vini*, *vinis*, *viparā*, *vipari*, *vipra*, *vyati*, *vyapa*, *vyabhi*, *vyava*, *vyā*, *vyut*, *saṃni*, *saṃpra*, *saṃprati*, *saṃpravi*, *saṃvi*, *sanni*, *sam*, *samanu*, *samabhi*, *samabhivi*, *samabhivyā*, *samava*, *samā*, *samut*, *samudā*, *samudvi*, *samupa*, *sampra*, *samprati*, *sampravi* [6].

Finally, the Invariable words are classified into firstly Indeclinable forms, such as adverbs, prepositions, conjunctions and other function words and particles listed in the lexicon, to which are added the root absolutives in *-tvā*, and secondly absolutive stems in *-ya*, necessarily prefixed by a non-empty preverb sequence.

---

[5] but fuller information about which *upasargas* are attested with which root is provided in the *Upasargārtha candrikā* by Cārudeva Śāstri (communicated by A. Kulkarni).

[6] in this list *tiras*, *puras* and *bahis* are not actual *upasargas*, but belong to the category of *gati* morphemes.

We see here how the distinction between the two forms of absolutives (called respectively *ktvā* and *lyap* in Pāṇinian terminology) is dealt with correctly by the regular grammar formalization.

The state graph shown in Figure 1 is only the super-structure at the phase level, since each phase lexical analysis uses a finer-grain state machine compiled from the corresponding lexical database. The inter-linking of the various machines is implicit from the operation of the reactive engine implementing the lexical analyser as a modular transducer [11]. We remark that proper sharing is induced by this mechanism. Thus the two phases Pv1 and Pv2 of the diagram correspond to a unique preverb sequences recognizer.

Since the writing of this paper for SALA, we augmented our transducer in order to deal with first level derivatives *kṛdantas* and *nan-samāsas*, leading to an analyser with 26 phases.

3 Preverb sandhi

Let us now discuss in greater detail the notion of "word" which is implicit from our state diagram. For the verb forms, we already mentioned that compound verbal forms, with auxiliaries *kṛ*, *as* and *bhū*, are accommodated, through the (productive) lexical category Iiv. The database Root contains all finite root forms, not just of primary conjugations, but of secondary conjugations as well when they are specified in the lexicon (causative, intensive and desiderative forms). Finally, the database Pv contains all preverb sequences listed above. All verbal forms are thus all words which are produced by transitions going from Verb to Accept in Fig. 1, arrows being interpreted by sandhi. Here one difficulty is looming, the fact that preverbs attach to root forms using ad-hoc sandhi rules, supplementing the usual external sandhi rules with rules triggering retroflexion of 's' (*niṣṭapati*, *niṣīdati*), *pariṣvajati*), of 'n' (*praṇakṣyati*, *nirṇayati*), euphonic 's' (*saṃskaroti*), elision of initial 's' (*uttabhnāti*), etc. Furthermore, such rules must operate through prefix operations of verbal morphology, such as augment for past or the various forms of reduplication of verbal stems. For instance, *samaskuruta* must be producible from affixing preverb *sam* to the imperfect form *akuruta* of root *kṛ*. Similarly, *pariṣaṣvaje* must be producible from affixing preverb *pari* to perfect *sasvaje* of root *svañj*. We currently have 24 special sandhi rules that handle this problem, and they seem to cover most cases. What is important is that they come in supplement to the usual rules, and thus do not hide them in standard sandhi, and moreover due to the modular nature of our segmenter they apply *only* to the transitions between Pv and Root, and thus do not provoke overgeneration in other sandhi situations.

Another difficulty is rather subtle, concerning preverb *ā*, which is special as being mono-phonemic. The problem arises form the fact that preverbs are not just morphemes, they are in a certain sense independent words, a remnant of their ancient role as postpositions in the Vedic age, where preverbs were not so rigidly attached to root forms as they became in the classical language. Thus preverbs must be thought of as combining by sandhi with the preceding word

in the sentence, prior to combining by sandhi with a root form. This is hardly noticeable, except in the case of preverb *ā*, which is so short as possibly causing a conflict between the two sandhi rules. Let us give an example. The imperative singular 2nd person of root *i*, to go, is *ihi* "go". Combined with preverb *ā*, indicating an action directed towards the locutor, it yields *ehi* "come". Now if we want to say "come here", we can precede this form with adverb *iha* "here". However, we do not get **ihaihi*, as the sandhi of *iha* and *ehi* would predict, but instead *ihehi*, obtainable by operating sandhi left to right, first between *iha* and *ā*, yielding *ihā*, and only then between this form and *ihi*. Because of the overlapping situation, sandhi left-to-right with autonomous *ā* gives a different result than directly glueing the adverb to the verb form. This exemple[7] shows that it would be wrong to generate a lexicon of verb forms where the preverbs would be affixed in advance. Furthermore, a special mechanism must be coined for verbal forms starting with *i* or *u* (short or long) in order to simulate the left-to-right sandhi for preverb *ā*. This mechanism (using the so-called *phantom phonemes*) has been explained in [8].

As we already said, there is a construction of compound verbal forms, using a special *ī* stem of nouns, compounded with verb forms of three auxiliary roots, *as*, *bhū* and *kṛ* (resp. to be, to become, and to do). This construction is explicit in the diagram of Fig. 1, with lexical categories Iiv (representing the special *ī* stems) and Auxi (representing the sub-lexicon of Root concerning the auxiliary forms).

Finally, there are verbs which are not obtained from roots, but from nominal stems, such as *śāntayati* "to appease" from *śānta* "peace". We treat these verbs more or less as roots, we assume they are lexicalized, even though their regular generation as verbs of present class 10 indicates that the process is fairly generative. Fortunately, it seems that this formation is rare from compound forms, like *amitrayati* "he is hostile", from *amitra* "enemy". But we note that such nominal verbs readily admit preverb affixing, like *anumantrayati* "he consacrates with a *mantra*", *ākarṇayati* "he listens", *parikhaṇḍayati* "he destroys" or *nirūpayati* "he determines".

4 Nominal morphology complexities

Let us now turn to substantival forms. An initial remark is that adjectives belong to the same morphological class as nouns, in contradistinction with later prakrits such as Hindi. Thus in Fig. 1 above, Subst will contain all primitive substantival forms, either nouns, adjectives, pronouns or numbers. The two databanks Noun and Ifc stand for inflected forms of nominal lexical items. Ifc contains bare root stems used as nouns, but usually restricted to right components of compounds. Their distinction from non-root nominal stems is essential for curbing overgeneration, since their stems are mostly mono-syllabic. We remark that the indication 'ifc' in standard dictionaries such as Monier-Williams' is ambiguous.

[7] already pointed out in: V. Henry. *Éléments de Sanscrit Classique*, Paris 1902.

It may mean either 'always used as right component of compound' (our use), or it may just prefix some particular sememe of a noun, which in this particular suffixal use has a specific meaning. The databank Iic contains bare stems of substantives, usable as left component of a compound. For instance, the entry *rājan* in the lexicon will generate (in Noun) all declined forms of *rājan*, for the various genders listed (here masculine) and for the various numbers and cases, plus (in Iic) the bare stem *rāja-* for compounds such as *rājapuruṣa*. Sometimes the lexicon indicates alternate Iic forms. For instance, *śvan* (dog) will generate alternates *śunas-* (for e.g. *śunaḥśepha* (dog's penis, a curious Brahmin name)) and *śva-* (for e.g. *śvavṛitti* (dog's life)).

The reader understands from Fig. 1 that we capture with our analyser the 'easy' cases of nominal compound forms, obtainable by the regular grammar:

$$\begin{aligned} \text{Subst} &:= \text{Noun} \mid \text{Compound} \\ \text{Compound} &:= \text{Iic} \cdot \text{Inflected} \\ \text{Inflected} &:= \text{Ifc} \mid \text{Subst} \end{aligned}$$

However, a discussion of the completeness of the approach is rather complex, since Sanskrit morphology is far from being an easy topic. We shall not discuss this issue in full detail, since it would require a very lengthy discussion, but will restrict ourselves to a few remarks, assuming familiarity with Gillon's classification of compounds [4].

One important remark is that our lexical analyser attempts to understand the structure of a compound as a (right recursive) linear structure of a sequence of components, rather than as a tree of dependencies. Thus the compound *ānandamayakośa* will be parsed as the (right associative) sequence *ānanda-(maya-kośa)* and *not* as the (left associative) dependency structure written as *(ānanda<maya)<kośa* in Gillon's notation, consistently with its compositional meaning (container of what makes joy). Similarly, a genuinely branching compound such as *avalokiteśvaraguṇakaraṇḍavyūha* will be parsed as *avalokita-īśvara-guṇa-karaṇḍa-vyūha* rather than the (more informative) tree structure *((avalokita<īśvara)<guṇa)<(karaṇḍa<vyūha)* (explanation of the qualities of *Avalokiteśvara*). This last exemple shows that we are gaining a reduction in complexity, since a list of $n+1$ items can be organised in C_n ways as a binary tree, where C_n, the n-th Catalan number, is exponential in n. Thus a compound with 7 components will get only one parse instead of 132. Of course the reverse side of the coin is that the tagging of compounds we shall get out of our segmentation process is definitely not sufficient to do the fine-grained analysis that is necessary to obtain the correct paraphrase of the compound, needed to understand its sense. However, it suffices for the *kāraka* analysis of the sentence.

The next consideration concerns the left component stem part. What is exactly stored in the Iic stem bank? I.e., what is exactly a "bare stem" ? The closest Pāṇinian notion corresponding to a Western linguistic notion of nominal stem is that of *pratipādikā*[8]. But this notion is both too rich and too poor. It is too rich,

[8] Properly, *pratipādikā* corresponds to nominal base, whereas *aṅga* corresponds to nominal stem (Cardona)

for instance in admitting compounding, whereas our first remark is to the effect that we do not need left recursion, parsing components from left to right, one at a time. It is too poor in that the so-called feminine stems used by Western grammars such as *durgā* are technically not *pratipādikās*, they are intermediate steps in the declension derivation, with a gender *pratyaya* (suffix). However, some of these feminine stems are productive as left components of compounds, such as *durgāpūjā*. Thus either the noun *durgā* (the Durgā divinity) is listed in the entry for adjective *durga* ('hard to access'), and then we must enter both stems *durga* and *durgā* in our Iic databank; or else noun *durgā* is listed separately in the lexicon as an autonomous entry, and then the lexicon must mark that indeed it is autonomous enough so that its stem *durgā* is admitted in the Iic databank. But we certainly do not wish to enter every feminine stem in Iic, there would be enormous overgeneration with most adjectives. Note that if *durgā* is listed separately from *durga* in the lexicon, there will be indeed ambiguity between forms of the adjective and forms of the deity name. And this is a very general problem with Sanskrit lexicography, since there is no special mark for proper names, and since furthermore all proper names derive from common nouns, we cannot make separate dictionaries for ordinary words and proper names, as is customary in Western languages. We are doomed to face the complexity of telling *Kṛṣṇa* from *kṛṣṇa* (black), since they are undistinguishable in *devanāgarī*.

Actually there are compounds, called *aluk* in the Pāṇinian terminology, which take a declined form as left component. Take for instance *ātmanepada*, whose left component *ātmane* is the dative of *ātman* ('for the self', said of middle voice verbal forms). Or *gavāmayana* ("cows' way" or year), where *gavām* is the genitive plural of *go*. Or also *yudhiṣṭhira* (steady in fight) where *yudhi* is the locative of *yudh*, with the additional peculiarity of enjoying retroflexion in its sandhi. Similarly *apsuyoni* (issued from waters) where *apsu* is the locative plural of *ap*. Here too, we must assume that this scheme is not productive, and all such compounds must be explicitly listed in the lexicon. Similarly for the *dvandva* compounds, where double dual forms are glued together, such as *mitrāvaruṇau*.

We have now covered the left components issued from substantives. Now we have to face other families of compounds, whose left component is not an autonomous nominal. Now we have two main families. What may occur as left component of a nominal compound, besides a nominal stem, is an indeclinable. We have already seen *aluk* compounds, whose left component was already declined. Now we deal with genuine indeclinables. Genuine indeclinables in Sanskrit fall in a number of categories. Some particles are function words that are not autonomous, like the coordination particles like *ca* (and) and *vā* (or), negations *na* and *mā*, and the quotation ending *iti*, a discourse operator. These do not enter compound formations, they operate at the functional level of linguistic analysis.

Then there are prepositions, like *pra*, which may be used as morphological prefixes, either as preverbs, or as compound-forming left components, the so-called *prādi tatpuruṣa*, in Gillon's classification, such as *pracaṇḍa* (terrible). The word *prādi* is a false example of such a *tatpuruṣa*, as being the compound of *pra* and nominal *ādi* (begins with), but *pra* is not analysed as the intensifying

preposition, but just quoted as the beginning of the list of prepositions in some contextual *gaṇapāṭha*. There is a standard didactic device in Pāṇinian terminology: you name a construct by a canonical representative, like '*tatpuruṣa*' (that person) is the canonical *tatpuruṣa* compound, but here *prādi* only looks like a *prādi*!

Many prepositions are actually syntactic postpositions, and are autonomous as such, especially in the Vedic language, such as *ā*, *prati*, etc. We already saw a vestigious remain of the postposition origin of *ā* in the sandhi analysis problem of *ihehi*. But in the classical language the combinatorics converged towards a prepositional usage, either as preverbs (our Pv databank in Fig. 1) or as left component of compounds. With often an interpretative dilemna. Should *prayoga* (effort) be analysed as a *kṛdanta* formation (1st level derivation of nouns from roots) of verb *prayuj* obtained by preverb *'pra'* prefixing root *'yuj'*, or as a *prādi tatpuruṣa* compound obtained by prefixing with preposition *'pra'* the *kṛdanta* derivative *'yoga'* of root *'yuj'* ? Sometimes the meaning commutes, and distinguishing the two would be redundant. But sometimes the distinction is anavoidable, like for *nirvācya*, which as a participle of verb *nirvac* means "what should be explained", whereas as a *prādi* compound it means "what should not be talked about".

We chose in our current implementation to make generative the preverb formation, uniformly for all known preverb sequences, but we chose to restrict *kṛdanta* forms to the ones obtainable from stems explicitly listed in the lexicon. This seems easier from a human user point of view, since he can refer to the meaning of a *kṛdanta* formation directly from the lexicon rather than computed through some hypothetical generic preposition composer. It has also the advantage of giving a finer grained description of the meaning when it is not compositional, which is often the case. We have actually developed a participles generator, which allows the segmentation to be complete over the full range of Sanskrit participles (past participle passive and active, present participle active middle and passive, future participle active middle and passive (of the 3 forms), perfect participle active and middle, in all three genders). We can then extend the segmenting automaton from the one shown in Fig. 1, and improve its recall. However this extended segmenter tends to over-generate, and the current thinking is to use it as a tool for incremental lexicon acquisition rather than as the main machinery.

All this digression may be summed up by the acknowledgement that in the current design, we do not analyse *prādi* compounds as such. This is true also of nominal prefixes such as *su* (good), *sa* (with), *dus*, *ku* (bad), *ko* and *vi* (ambiguous between not and very). Thus the compound *vimukha* will be recognized whole as a lexical item, although in the lexicon its etymology is recognized as *vimukha*. And other similar compounds may fail to be recognized, if they have not been lexicalised. Note that often retroflexion arises at the junction with these prefix morphemes, like in *durṇāman* and *viṣama*, which justifies treating this

construction within derivational morphology rather than compounding.[9] But after all, look at Monier-Williams, and you will see that there is no unique entry for the prefix *su*, that would run for many pages. A half measure compromise was made, by distributing all *svādi* compounds along with the first letter of their right component. The same goes for *sa*.

We have kept for the end the most common and vicious particle, namely the privative prefix *a-* (*an-* in front of vowels). It forms the so-called *nañ tatpuruṣa* class [4]. The mind boggles at the idea that phoneme *a* could be treated as autonomous for segmentation. We already have the problem of *ā* splitting four different possible ways occurring often enough without having a generative privative prefix, so we decided here too not to treat *nañ* constructions as compound forms, but just as internal morphology of lexicalised items. After all, if the meaning of this particle was compositional, the Pune dictionary project could be finished very soon, once they complete the dictionary entries beginning with *an*. We could just get the meaning of any noun by looking up its hypothetical *nañ* compound and inverting its meaning!

We are not quite done yet with *aluk* compounds. Some compounds have an indeclinable as left component. We find adverbials such as the coordinations *yathā* and its correlative *tathā* (thus), but also the odd list *mithyā*, *satrā*, *sadha*, *saha*, *punar*, *upari*, *alam*. Now there are several cases. Sometimes these adverbs form compounds such as *tathāgata* (thus come), which keep the nominal category of their right component, and decline in the same way. There is no reason to treat these compounds as different from the standard *tatpuruṣas*. Then either we could make them generative, by entering say *tathā* in the Iic bank, or else just keep whole the ones that are lexicalised, and avoid overgeneration with their autonomous use. It is a design choice.

But there is also the case of so-called *avyayībhāva* compounds "turned into indeclinables". The whole compound now inherits its adverbial role from its left component, and it becomes an invariable form itself. The problem is, what is the assumed form of the right component, with respect to its stem ? Well, it depends. Sometimes, it is a declined form, typically a singular neuter in the accusative. For instance, consider *yathāsthānam* (each one at its proper place). It can be analysed similarly to a *tatpuruṣa*, glueing the Iic component *yathā* to the accusative declension *sthānam* of its right stem *sthāna*. And it will be left to the syntax to recognise this accusative as an adverbial, a quite common construction. The fact that the hypothetical *yathāsthāna* is used only in this form is only in the eyes of the beholder, there is nothing special here. Indeed our lexer may recognize separately *yathā* as an adverb and *sthānam* as the accusative form of a neuter substantive used as an adverbial, and leave it to a further analysis to glue the two adverbs into an *avyayībhāva* compound.

[9] Let us note that it may even happen that the right hand side of such a compound be a verbal form, like in *svasti*, where *su* is glued to the present form *asti* (it is); another irregular compound is *naciketa*, built by compressing a verbal phrase in the passive "(he who) did not understand". Such irregular constructions are clearly not productive.

However, there is still another situation, which I shall call *genuine avyayībhāva*. It is when the right component is not a *bona fide* form of the right stem. For instance, *yathāśakti* (as much as possible), where *śakti* is the bare stem of feminine noun *śakti*, and thus not one of its declensions. Another example is *yathāśraddham* (according to your convictions), where the right component is similar to the accusative form of a hypothetical neuter stem *śraddha* induced from the original feminine stem *śraddhā*. In both of these cases, there is no hope of parsing the compound as an initial Iic form *yathā* followed by a form of the right stem *śraddhā*. It is clear then that we have to augment our phase automaton with another path for indeclinables thus constructed. Again there will be a design decision concerning the trade-off between a generative scheme or a mere lexicon lookup for lexicalised genuine *avyayībhāvas* considered as exceptions.

We are almost done with compounds. If we look at Gillon's classification, we have not yet discussed *dvandva* compounds, which pose no special problem, except that some are *aluk*, like double duals such as *mitrāvaruṇau*, but such constructions are probably not that productive in the modern language, and Vedic forms may be lexicalised. The case of so-called *upapada tatpuruṣa* concerns the part Ifc of Fig. 1 which we already discussed. So only one family is left, the problematic *bahuvrīhi* ("much riced", i.e. rich).

First of all, let us mention that we do not consider *bahuvrīhi* a separate morphological class of compounds, but a specific usage of certain compounds, namely their exocentric usage (*anyapadapradhāna*) as an adjective. Almost any *tatpuruṣa* is liable to be used as *bahuvrīhi*, which will be undistinguishable from it, at least if accent is not marked. For instance, most *prādi* compounds, such as *pramukha* (principal), are usable as exocentric compounds, and thus the exocentric usage is orthogonal to the morphological classification. For instance, the nominal phrase *bahuvrīhiḥ puruṣaḥ* (rich man) does not pose any specific problem. There is concord in gender, number and case, and this nominal phrase is unproblematic. The problem arises if we want to use this adjective at a different gender, for instance for a rich woman, since *vrīhiḥ*, rice, a masculine substantive, has only masculine declensions to offer, and thus we cannot obtain say the analogue *bahuvrīhyā striyā* "by the rich woman" since *vrīhyā* is not a form of *vrīhi*. Similarly *caturmukhaḥ* (he who has four heads) needs the masculine form *mukhaḥ* that the neuter substantive *mukha* does not produce. This problem is probably the biggest challenge for segmentation, since exocentric usage is quite common. Here again we have a design decision to take. Either we create a new lexical category of multi-gendered nominals to operate as right components of *bahuvrīhis* - with the obvious risk of overgeneration, since now many forms will be ambiguous as adjectives or as nouns, introducing a new potential exponential explosion. Or else we limit ourselves to *bahuvrīhis* recognized as such in the lexicon, for which we produce all necessary forms. This is the course followed at present, with automated recording of compounds which possess more genders than their right component.

Now that we have finally examined all manners of compounds, and made sure that we have derivational morphology processes that are sufficient for gen-

erating their components, we should still check that our sandhi splitter will split them indeed, i.e. that their forms are obtained by glueing their components with external sandhi. This is generally the case, but there are certain exceptions, specially in proper names, such as *Rāmāyaṇa*, whose assembly uses retroflexion, and which is therefore not obtainable by external sandhi from its components *rāma* and *āyana*. We already noted the case of *Yudhiṣṭhira*, which is *aluk* in addition. Such exceptions may be collected systematically from the lexicon, so that such irregular compounds are stored whole. Proper names ought to be lexicalized whole, anyway.

A remark is in order here. We consider *aluk* compounds as *a priori* exceptions, that is we do not attempt to generate as compounds all compositions of *padas* (fully declined words). This is different in spirit to the Pāṇinian methodology, where compounds are generated as composition of *padas*, except that the suffix potentially transforming its left hand side stem into a *pada* is dropped for all compounds except the *aluk* ones. This methodology would be disastrous for a mechanical analysis, since it would potentially generate a non-determinism branching factor of 21 (all the morphological forms of a gendered substantival stem under 3 numbers and 7 cases, and even 3 times worse for an adjective having potentially 3 genders) instead of considering only one solution with the bare stem, except in a few exceptions recorded in tables. Even in these cases we take only one form, not all possible ones. This discussion shows that the choice of following a Pāṇinian process or not is not a matter of taste, but may be justified by the specific problem addressed. Pāṇini's method is the natural one for generation, since it assumes that one starts from a semantically meaningful utterance, and ends up with a justification for its phonetic realisation. Thus e.g. the *aluk* compound *apāṃnapāt* glues the genitive plural *apām* to its right hand side *napāt* justifiably because it designates *Agni* as "issued from the waters", and similarly the non-*aluk* compound *rājapuruṣa* is obtainable by contraction of its paraphrase *rājñaḥ-puruṣa* as a *ṣaṣṭatpuruṣa* "servant of the king". This makes perfect sense for constructing semantically meaningful sentences. But we are in a completely different game when the computer tries to analyse a sentence without having the slightest idea of whether it makes sense or not.

We made a very long story about segmenting compounds, showing that the problem is far from trivial. The problem is to define generative morphology in regular terms, so that it is amenable to description by finite-state processes. One difficulty is the non-regular nature of retroflexion processes of internal sandhi. Another difficulty is the highly recursive nature of Sanskrit generative morphology, as witnessed for instance in the word-formation diagram in [2]. Roots produce nominals in two rounds (*kṛdanta* then *taddhita*), which combine in compounds, but these may in turn produce nominal verbs which will iterate the whole process. Verbal compounding too is productive. After all, the word *avyayībhāva* itself is a *kṛdanta* of compound verb *avyayībhū*. If we want to capture analytically such forms, we shall have to add transitions from the Iiv prefix bank to a special bank of nominal derivatives of auxiliary verbs, etc. As usual, the more complete the derivational morphology we implement, in order to improve recall,

the more overgeneration we shall get, degrading precision to unacceptable levels. Thus some compromise must be achieved between generative and lexicalised information, so that we obtain useful parsers.

Another issue that is more or less hidden in morphology production is the assumption that stems are made up of strings of phonemes. This assumption is implicit in Pāṇini's grammar, since phonetico-morphological markers such as *pratyāhāras* use actual Sanskrit phonemes as encodings in a slick manner, in order to prevent confusion between these markers and actual phonemic segments. In the end of the morphological process all markers are erased, but it is not clear that all intermediate stems may be purged of their markers. And indeed a great freedom exists for the naming of the roots, which are not really autonomous, whence the wide discrepancies in the nomenclature of roots in the various grammars. Thus the root called *hū* by Renou is called *hū* or *hvā* by Whitney, and *hve* by Burnouf and Gonda, whereas in the Pāṇinian tradition it is referred to as *hve*, but the stem *hū* is called its *aṅgam*. In Western presentations of Sanskrit, it is more or less assumed that one may present morphology in such a way that all stems are genuine strings of phonemes, at the expense of multiplying the paradigms. Thus it is assumed that a nominal stem has enough information to point to its generative paradigm (some function of its morphological parameters gender, number and case). But this is only partially true. For instance, you cannot tell what is the nominative singular of a masculine stem in *-vas* without knowing whether it is obtained as a perfect participle (like *vidvān*) or not (like *ravāḥ*). This difficulty is usually swept under the rug, but the proper informational modeling requires solving rigorously such problems.

5 Filtering out nonsensical segmentations

Now that we have explained our segmentation process, we may put it to actual use and test it. The whole implementation is rather complex. First of all, a generative lexicon must be set up in order to collect the morphological parameters of the roots and other autonomous items. Then the banks of forms, sorted out by lexical categories, must be mechanically constructed. Then these banks of forms are compiled into sandhi splitting transducers, using the technology explained in [7]. Once all these static preparatory steps are finished, we may start the actual segmentation of Sanskrit enonciations, by preprocessing the stream of transliterated input into independent chunks, which are then sent to the segmentation process proper, which will find all sandhi analyses and report them as a stream of potential solutions. Each solution is listed as a sequence of *bona fide* morphological segments, interleaved with sandhi indications. Each segment may be optionally tagged with the morphological parameters, i.e. presented as a lemmatised stem, where the stem points to the lexicon. Thus we may inspect every returned potential solution and reconstruct a full generation.

A too naive implementation of this process, even using a simplified lexicon[10], is not directly usable because of overgeneration. There are literally thousands of potential solutions, even for small sentences (typically hemistisch or half *ślokas*). A typical ambiguity is between vocatives and compound-formation of masculine stems in -*a*, leading to exponential behaviour. This is remedied by demanding vocatives to be isolated in separate chunks. This is consistent with the fact that they are not really part of the sentence, but are really discourse components, and that they are distinguished by prosody, so that sandhi is limited. Thus vocative forms are not mixed up with the Noun database of genuine declensions, and are kept in their own separate lexical databank.

Even if one evacuates vocatives and other interjections, there are far too many possible segmentations, most of them not genuine ambiguities like we saw in the *śvetodhāvati* example, but just totally nonsensical assemblage of words. Thus it is absolutely mandatory to filter out of the stream of segmentations the solutions which do not make sense from a syntactico-semantical point of view. We have implemented a constraint programming algorithm to effect such filtering, which is sketchily described in [9]. This algorithm amounts to some kind of semantic role analysis, similar to the *kāraka* analysis of *Pāṇini*. It differs from *Pāṇini*'s analysis in that our analysis is performed on the morphological parameters (cases mediated by voice and mood) rather than on semantic roles. For instance, in the active voice, it is the subject in the nominative that bears the agent role, as opposed to the verbal *vibhakti*. We believe that the main interest of *Pāṇini*'s *kāraka* model is related to the pro-drop character of Sanskrit, where often the agent is altogether ellipsed because it is understood from the context as the topic or theme. On the other hand, a more traditional subject-predicate modeling has the advantage that the concord of a verb and its subject is prominent in the dependency matching, instead of being relegated to an auxiliary role.

We believe that this semantic consistency checking ought to operate at the level of discourse (*mahāvākya*) rather than simple sentences (*vākya*), so that ellipsed elements may be recovered from the discourse context as their anaphoric antecedents. Thus a proper constraint-based segmenter should progressively build a context of semantic roles, in order to construct a dependency structure correctly scoped in this context. We also believe that the proper treatment of mutual coercions between adjectives and substantives ought to emerge from a process of co-indexation, as suggested by Kiparsky [13].

This software is regularly released as a set of Web services publically accessible from the site `http://sanskrit.inria.fr/` where facilities are given to select from filtered solutions the intended interpretation. A lot remains to be done to treat correctly coordination and relative phrases. The context management is only in its design phase. The user may specify a contextual topic to help for ellipse management, this is mostly for testing purposes. The segmenter has been tested on simple exemples, taken from an infant primer [14], from sim-

[10] currently our lexicon has 18565 entries, that generate roughly some 250000 nominal forms and 140000 finite root forms; the experimental extended system generates more than 300000 participial forms.

ple *subhāṣita* sentences (proverbs), and from the examples provided in Apte's grammar [1]. For the last family, we gratefully acknowledge the cooperation of Brendan Gillon, who put at our disposal his data base of formal analyses of such. Our current goal is to converge on a tagging prototype conceived of as an interactive tool for the user to analyse a piece of Sanskrit text, select the intended interpretation, and build a functional structure exhibiting proper dependencies, with co-indexation and anaphora links in a discourse structure. Once this facility is operational, it will be much faster to analyse a significant number of characteristic sentences, such as the full set of Apte's examples, the examples given in commentaries to *Pāṇini*'s grammar, and some simple texts like *Nala*'s story, for which e.g. Lanmann's reader provides enough explanations to ensure correct analysis. In parallel, the generative machinery for participles could be extended to a fuller *kṛdanta* generator. Also some extension of the Noun forms generator could allow for generative *taddhita* formations, such as the quality nominals in *tā* or *tva*. This would facilitate lexicon acquisition from the corpus. The result of this development would be the construction of a Sanskrit discourse treebank, available for optimisation. If the treebank is large enough, it will be possible to use statistical methods to train the constraint evaluation mechanism and optimise its discriminative power, in order to improve the precision of the tagger. It is hoped that we may obtain a tagger precise enough that the non-determinism will be reduced to genuine ambiguity resolution, and thus that the semi-automatic tool will be progressively improved into a fully deterministic analyser applicable to real corpus. We thus hope to bridge the gap with the statistical tagger of Hellwig [6, 5], whose methods ought to apply to our optimisation problem.

Acknowledgment. A preliminary version of this paper was presented in october 2009 at the XXVIII SALA (South Asia Languages Analysis) Roundtable in Denton, Texas.

References

1. V. S. Apte. *The Student's Guide to Sanskrit Composition. A Treatise on Sanskrit Syntax for Use of Schools and Colleges.* Lokasamgraha Press, Poona, India, 1885.
2. A. Bharati, A. Kulkarni, and V. Sheeba. Building a wide coverage Sanskrit morphological analyser: A practical approach. *Private communication*, 2008.
3. B. S. Gillon. Autonomy of word formation: evidence from Classical Sanskrit. *Indian Linguistics, 56 (1-4)*, pages 15–52, 1995.
4. B. S. Gillon. Tagging classical Sanskrit compounds. In A. Kulkarni and G. Huet, editors, *Sanskrit Computational Linguistics 3*, pages 98–105. Springer-Verlag LNAI 5406, 2009.
5. O. Hellwig. Extracting dependency trees from Sanskrit texts. In A. Kulkarni and G. Huet, editors, *Sanskrit Computational Linguistics 3*, pages 106–115. Springer-Verlag LNAI 5406, 2009.
6. O. Hellwig. SanskritTagger, a stochastic lexical and POS tagger for Sanskrit. In G. Huet, A. Kulkarni, and P. Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 266–277. Springer-Verlag LNAI 5402, 2009.
7. G. Huet. A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger. *J. Functional Programming*, 15,4:573–614, 2005.

8. G. Huet. *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, 2006.
9. G. Huet. 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, 2007. ACM.
10. G. Huet. Formal structure of Sanskrit text: Requirements analysis for a mechanical Sanskrit processor. In G. Huet, A. Kulkarni, and P. Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*. Springer-Verlag LNAI 5402, 2009.
11. G. Huet and B. Razet. The reactive engine for modular transducers. In K. Futatsugi, J.-P. Jouannaud, and J. Meseguer, editors, *Algebra, Meaning and Computation, Essays Dedicated to Joseph A. Goguen on the Occasion of His 65th Birthday*, pages 355–374. Springer-Verlag LNCS vol. 4060, 2006.
12. P. P. Joshi. A glimpse into the *apadam*-constraint in the tradition of Sanskrit grammar. In G. Huet, A. Kulkarni, and P. Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*, pages 278–286. Springer-Verlag LNAI 5402, 2009.
13. P. Kiparsky. On the architecture of Pāṇini's grammar. In G. Huet, A. Kulkarni, and P. Scharf, editors, *Sanskrit Computational Linguistics 1 & 2*. Springer-Verlag LNAI 5402, 2009.
14. V. Sastri. *Samskrita Bālādarśa*. Vadhyar, Palghat, 2002.
15. J. S. Speijer. *Sanskrit Syntax*. E. J. Brill, Leyden, 1886.