Sanskrit and Computational Linguistics

Akshar Bharati,
Amba Kulkarni
Department of Sanskrit Studies
University of Hyderabad
Hyderabad
apksh@uohyd.ernet.in

30th Oct 2007

1 Introduction

How a language communicates information intrigued Indian thinkers since millenia. This led to different theories of language analysis. *Pāṇṇini*s grammar saw the culmination of different thoughts into his monumental work *ashtādhyāyi*. The modern age of information theory has provided a new boost to the studies of *ashtādhyāyī* from the perspective of information coding.

The importance of *ashtādhyayī* is three fold. The first one, as is well known, as an almost exhaustive grammar for any natural language with meticulous details yet small enough to memorize. Though *ashtādhyāyī* is written to describe the then prevalent Sanskrit language, it provides a grammatical framework which is general enough to analyse other languages as well. This makes the study of *ashtādhyāyī* from the point of view of concepts it uses for language analysis important. The third aspect of *ashtādhyāyī* is its organization. The set of less than 4000 *sūtra*s is similar to any computer program with one major difference the program being written for a human being and not for a machine thereby allowing some non-formal or semi-formal *sūtra*s which require a human being to interpret and implement them. Nevertheless, we believe that the study of *ashtādhyāyī* from programming point of view may lead to a new programming paradigm because of its rich structure. Possibly these are the reasons, why Gerard Huet feels that Panini should be called as the father of informatics[1].

The Indian grammatical tradition with three schools of *shābdabodha* viz. *vyākaraṇa*, *nyāya*, and *mīmānsā* offer various levels of linguistic analysis which is directly relevant to computational linguistics.

[1] Inaugural speech at the First International Sanskrit Computational Symposium, 2007.

Apart from the *ashtādhyāyī* and the grammatical tradition, the rich knowledge base in Sanskrit has been a source of attraction for both Indian as well as western scholars. Sanskrit was at one time ”Lingua Franca” of the world of intellectuals, in addition to being a spoken language. As such, we find Sanskrit rich with many scholarly texts in different disciplines of studies – ranging from Astronomy, *Āyurveda* to different schools of Philosophy. Computational Linguistics can play a major role in developing appropriate tools for Sanskrit, so that this rich knowledge can become available to the interested scholars easily.

Thus both Sanskrit and Computational Linguistics have a lot to offer to each other. Akshar Bharati group has been engaged in both the tasks viz,. the task of developing computational tools for Sanskrit as well as the task of using Indian Grammatical thought for the analysis of other Indian Languages.

We first give a brief sketch of Akshar Bharati et al's work in the area of Sanskrit for Computational Linguistics followed by its work in the area of Computational Linguistics for Sanskrit.

2 Sanskrit for Computational Linguistics

2.1 Theoretical Aspect

It is believed by many scholars that though *Pāṇini* has written a grammar for Sanskrit, the concepts he used are general ones and thus it providesa framework to write grammars for other languages. As a first step towards applying *Pāṇini*an grammar, a parser based on *Pāṇini*an Grammar formalism was developed to analyse Hindi sentences. This parser based on *kāraka* theory used Integer Programming to analyse simple Hindi sentences.(Bharati, 1994) A tagged corpus for Indian languages is also being developed based on *Pāṇini*an Grammar, at LTRC, IIIT, Hyderabad.

*Pāṇini*an Grammar gives utmost importance to the ”information” coded in a language string. The *svatantraH kartā*(P 1.4.54) of *Pāṇini* establishes the fact that what can be extracted from a language string are the only *kāraka* relations, and not the thematic roles. To extract thematic roles, one needs to appeal to the world knowledge. But what a language codes through its coding scheme is only the *kāraka*s.
E.g. consider the sentences

rāmaH tālam udghāṭayati.
kuncikā tālam udghāṭayati.
tālaHudghāṭyate.

We see that in all these sentences, *rāmaH*, *kunchikā* and the *tālaH* are the *kartā*s of the verbs *udghāt*, whereas they do not have the same thematic roles.

To assign the thematic roles, one needs to appeal to the general knowledge.

Akshar Bharati group is also looking at English from *Pāṇini*an perspective(Bharati,2005). It was observed that the major difference between English and Indian languages say, e.g. Hindi, are that English does not have an overt accusative marker at phonemic level and also there is no morpheme in English corresponding to yes-no question marker in Hindi. The information loss caused by the absence of phonemic level accusative marker is compensated by the sacroscency of Subject position in English. This brings in several structural differences between Hindi and English which have been illustrated in the figure 1.

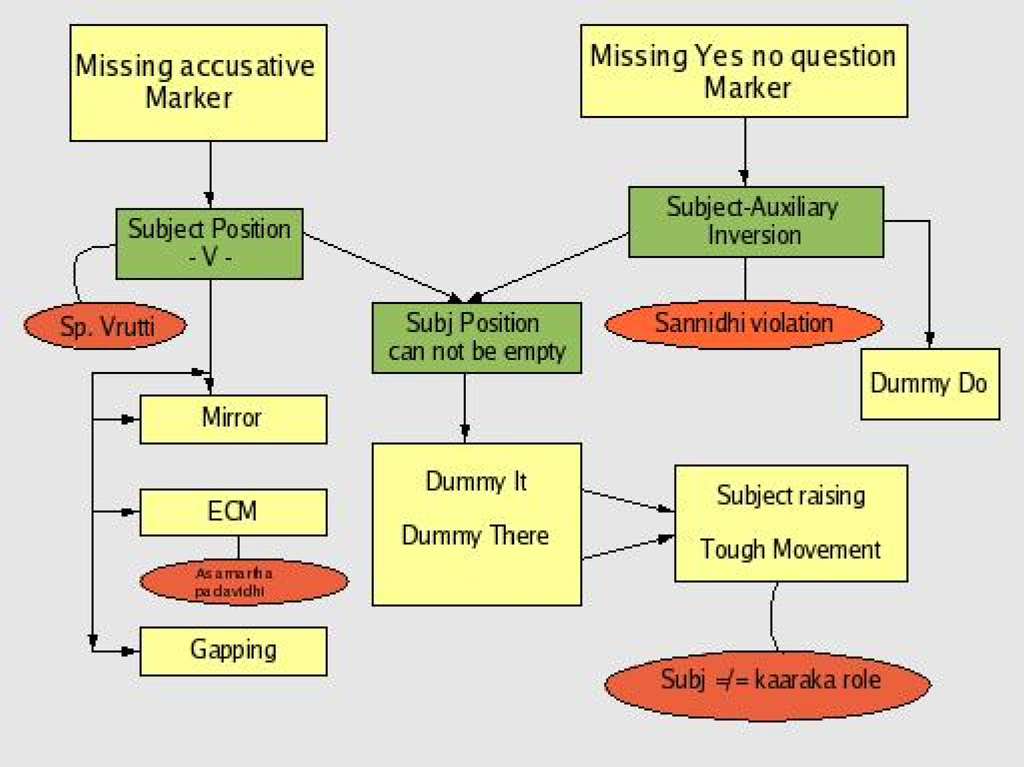


Figure 1: Structural contrast between English and Hindi

2.2 Practical Aspect

On practical side of this, Akshar Bharati has developed anusārakas a language accessors, to cater to the needs of people who want to acess material in languages unknown to them.

Translation involves not only transfer of content but also creativity in expressing the source language content into target language. In general any two languages are incommensurate in their expressions. This makes translation lack faithfulness. In other words, there is always a tension between faithfulness and naturalness. If one wants to ascribe to faithfulness to the content in the source language(SL), one has naturally to give up the naturalness or beauty of the target language(TL). The moment one tries to translate a SL text to sound natural in TL, some factor of 'unfaithful'-ness to the SL creeps in.

Therefore, if one is interested in reading some serious texts such as texts on laws or some scientific texts, one would not like to depend upon the translation but rather one would like to look at the original texts and interpret them on his/her own. To look at the original texts, then one should know the source language perfectly. However, this is not an easy task. The question is - can computers help a serious reader, say of Sanskrit, to understand the Sanskrit texts with the help of computers?

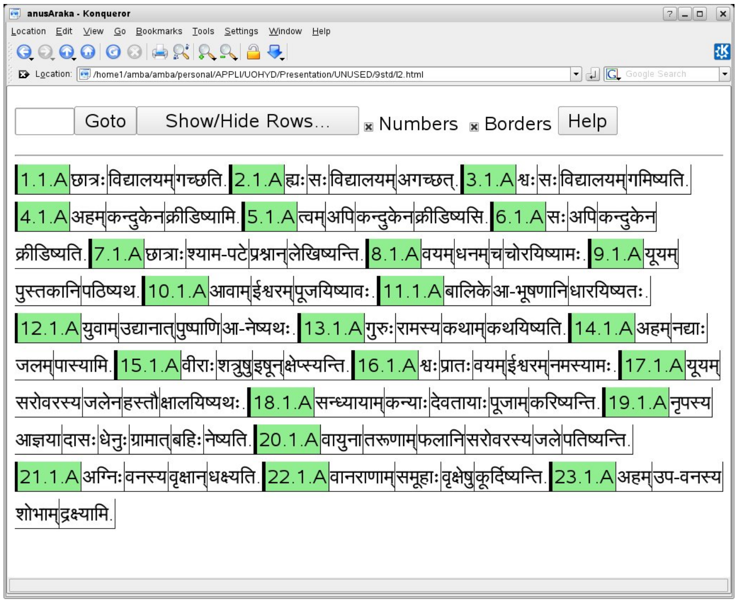
Anusāraka or language accessor attempts to provide such a help to the serious reader. It distinguishes between the reliable sources of information from the heuristic sources. The output is generated in layers with the topmost layer producing the image of the source text and other layers providing the graded output leading to Machine Translation.

Our major effort is in the area of English-Hindi anusāraka system(Kulkarni, 2003). However, over the past few years, we have also looked at Sanskrit-Hindi pair and some prototypes are available for demonstration. Unlike Machine Translation(MT), anusāraka is aimed as a language accessor and not a MT system giving just final translation. Here the user is an important and integral part of the system. Anusāraka differs from MT in two different ways:

- The architecture of anusāraka ensures that modules with high reliability are used before the modules with less reliable outputs, thereby ensuring maximum benefits at the early modules. To avoid the cascading of intermediate outputs leading to more unreliability, output at each level is made available to the user.
- An intelligent user interface not only allows the user to hide the undesired information but also provides context based help. The advantage of this interface is that the user has full control over the interface and thus he/she can display only the information of his/her own choice curtailing or hiding the other information.

In what follows we explain in brief the working of Sanskrit-Hindi anusāraka with sample screen shots of Sanskrit-Hindi anusāraka outputs.

1. Fig 2 contains screen shot of simple Sanskrit sentences, delimited by green strip, marking boundary of a sentence. Each word is being shown in different cell of a table. It is assumed that the words have been split manually before feeding the text to anusāraka.


anusAraka - Konqueror
Location Edit View Go Bookmarks Tools Settings Window Help
Location: /home1/amba/amba/personal/APPLI/UOHYD/Presentation/UNUSED/9std/l2.html
Google Search
Goto
Show/Hide Rows...
☒ Numbers
☒ Borders
Help

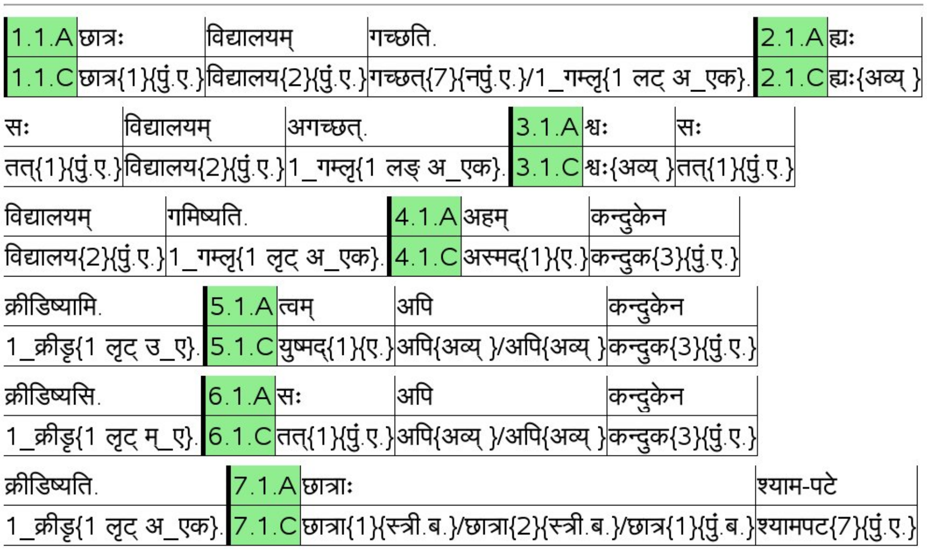
1.1.A छात्रः विद्यालयम् गच्छति. 2.1.A ह्यः सः विद्यालयम् अगच्छत्. 3.1.A श्वः सः विद्यालयम् गमिष्यति.
4.1.A अहम् कन्दुकेन क्रीडिष्यामि. 5.1.A त्वम् अपि कन्दुकेन क्रीडिष्यसि. 6.1.A सः अपि कन्दुकेन
क्रीडिष्यति. 7.1.A छात्राः श्याम-पटे प्रश्नान् लेखिष्यन्ति. 8.1.A वयम् धनम् च चोरयिष्यामः. 9.1.A यूयम्
पुस्तकानि पठिष्यथ. 10.1.A आवाम् ईश्वरम् पूजयिष्यावः. 11.1.A बालिके आ-भूषणानि धारयिष्यतः.
12.1.A युवाम् उद्यानात् पुष्पाणि आ-नेष्यथः. 13.1.A गुरुः रामस्य कथाम् कथयिष्यति. 14.1.A अहम् नद्याः
जलम् पास्यामि. 15.1.A वीराः शत्रुषु इषून् क्षेप्स्यन्ति. 16.1.A श्वः प्रातः वयम् ईश्वरम् नमस्यामः. 17.1.A यूयम्
सरोवरस्य जलेन हस्तौ क्षालयिष्यथः. 18.1.A सन्ध्यायाम् कन्याः देवतायाः पूजाम् करिष्यन्ति. 19.1.A नृपस्य
आज्ञया दासः धेनुः ग्रामात् बहिः नेष्यति. 20.1.A वायुना तरूणाम् फलानि सरोवरस्य जले पतिष्यन्ति.
21.1.A अग्निः वनस्य वृक्षान् धक्ष्यति. 22.1.A वानराणाम् समूहाः वृक्षेषु कूर्दिष्यन्ति. 23.1.A अहम् उप-वनस्य
शोभाम् द्रक्ष्यामि.

Figure 2: Sanskrit text – with pada pātha

2. Fig 3 contains morphological analysis corresponding to each word. For example the morphological analysis of the word *chātra.h* is *chātra* {1} {pu. e. },
   where *chātra* is the *prātipadika*,
   1 – vibhkati {case}
   pu – gender -masculine
   e. – number - singular

   Similarly,
   gamlṙ {1 lat a-eka}
   stands for
   dhātu: gamlṙ
   1 –> gaṅa
   lat –> lakāra
   a –> person - 3rd person
   eka –> number - singular

   This kind of analysis is useful for anybody who has some basic knowledge



1.1.A	छात्रः	विद्यालयम्	गच्छति.	2.1.A	ह्यः
1.1.C	छात्र{1}{पुं.ए.}	विद्यालय{2}{पुं.ए.}	गच्छत्{7}{नपुं.ए.}/1_गम्लृ{1 लट् अ_एक}.	2.1.C	ह्यः{अव्य् }

सः	विद्यालयम्	अगच्छत्.	3.1.A	श्वः	सः
तत्{1}{पुं.ए.}	विद्यालय{2}{पुं.ए.}	1_गम्लृ{1 लङ् अ_एक}.	3.1.C	श्वः{अव्य् }	तत्{1}{पुं.ए.}

विद्यालयम्	गमिष्यति.	4.1.A	अहम्	कन्दुकेन
विद्यालय{2}{पुं.ए.}	1_गम्लृ{1 लृट् अ_एक}.	4.1.C	अस्मद्{1}{ए.}	कन्दुक{3}{पुं.ए.}

क्रीडिष्यामि.	5.1.A	त्वम्	अपि	कन्दुकेन
1_क्रीडृ{1 लृट् उ_ए}.	5.1.C	युष्मद्{1}{ए.}	अपि{अव्य् }/अपि{अव्य् }	कन्दुक{3}{पुं.ए.}

क्रीडिष्यसि.	6.1.A	सः	अपि	कन्दुकेन
1_क्रीडृ{1 लृट् म_ए}.	6.1.C	तत्{1}{पुं.ए.}	अपि{अव्य् }/अपि{अव्य् }	कन्दुक{3}{पुं.ए.}

क्रीडिष्यति.	7.1.A	छात्राः	श्याम-पटे
1_क्रीडृ{1 लृट् अ_एक}.	7.1.C	छात्रा{1}{स्त्री.ब.}/छात्रा{2}{स्त्री.ब.}/छात्र{1}{पुं.ब.}	श्यामपट{7}{पुं.ए.}

Figure 3: Sanskrit text – with morph analysis

of Sanskrit morphological analysis, and has a good vocabulary of Sanskrit. For example, any Indian with good knowledge of mother tongue, and some background of Sanskrit should find this layer of immense use.

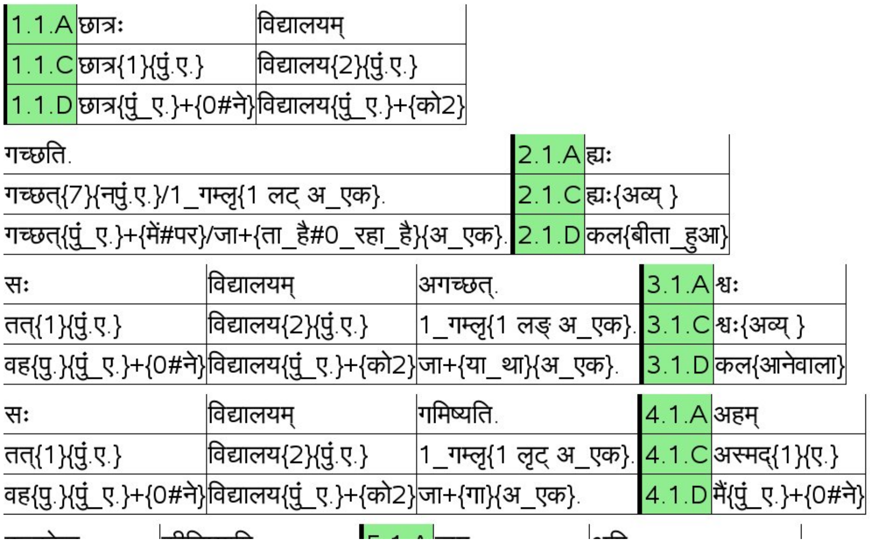
In this step, as one can see, if there are multiple morphological analysis possible, all the answers for each word are being displayed. It is the context that decides which one is the correct answer. At this stage, machine does not take any decision, since a morphological analyser analyses single word at a time.

One can then think of another layer, similar to a Part of Speech tagger, where machine uses some heuristic rules to rule out undesirable answers. Of course, the reliability of this layer can not be 100%.

One can think of several such layers, starting from marking say correct morphological Analysis to grouping the visheshya-visheshanas, local word groupings such as 'rāmeṇa saha', 'gacchati sma' etc., to kāraka analysis and sharing of kārakas - that requires a full fledged parser, and finally a Word Sense Disambiguation module. However, at each stage the reliability of the system goes down and the cascading effect will further reduce the reliability of translation.

3. Figure 4 provides Hindi gloss for each word with separate glosses for the root and the suffix. The meaning of a word is then composed by the word

generation module. In case the meaning is non compositional, it is directly provided in the dictionary as an exception.



1.1.A	छात्रः	विद्यालयम्
1.1.C	छात्र{1}{पुं.ए.}	विद्यालय{2}{पुं.ए.}
1.1.D	छात्र{पुं_ए.}+{0#ने}	विद्यालय{पुं_ए.}+{को2}

गच्छति.	2.1.A	ह्यः
गच्छत{7}{नपुं.ए.}/1_गम्लृ{1 लट् अ_एक}.	2.1.C	ह्यः{अव्य् }
गच्छत{पुं_ए.}+{में#पर}/जा+{ता_है#0_रहा_है}{अ_एक}.	2.1.D	कल{बीता_हुआ}

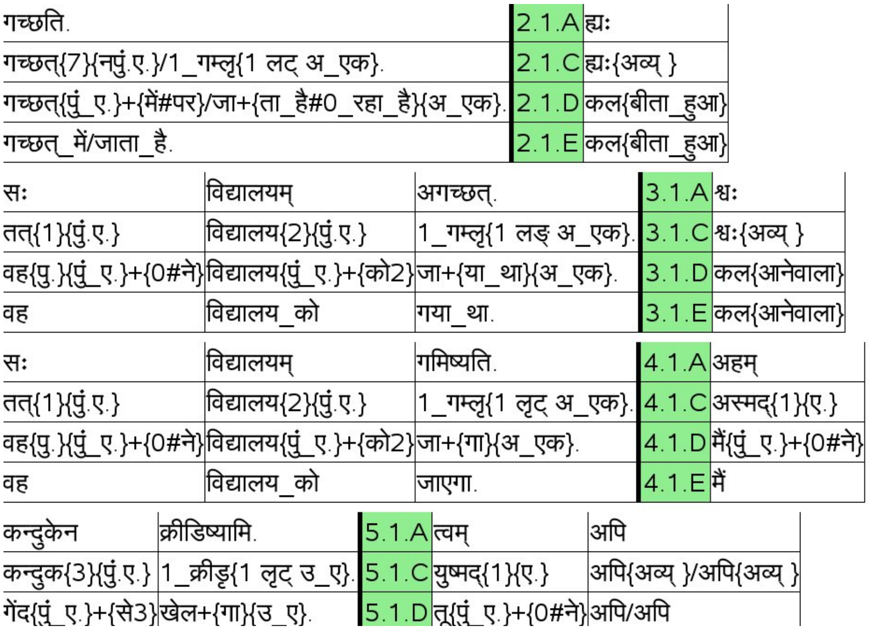
सः	विद्यालयम्	अगच्छत्.	3.1.A	श्वः
तत्{1}{पुं.ए.}	विद्यालय{2}{पुं.ए.}	1_गम्लृ{1 लङ् अ_एक}.	3.1.C	श्वः{अव्य् }
वह{पु.}{पुं_ए.}+{0#ने}	विद्यालय{पुं_ए.}+{को2}	जा+{या_था}{अ_एक}.	3.1.D	कल{आनेवाला}

सः	विद्यालयम्	गमिष्यति.	4.1.A	अहम्
तत्{1}{पुं.ए.}	विद्यालय{2}{पुं.ए.}	1_गम्लृ{1 लृट् अ_एक}.	4.1.C	अस्मद्{1}{ए.}
वह{पु.}{पुं_ए.}+{0#ने}	विद्यालय{पुं_ए.}+{को2}	जा+{गा}{अ_एक}.	4.1.D	मैं{पुं_ए.}+{0#ने}

Figure 4: Sanskrit text – with hindi meaning

4. If a parser exists, the next layer (see fig 5) takes care of agreement and generates the Hindi or target language output. In the present case, since the parser for Sanskrit does not exist, the generated output lacks agrement information, thereby making the output ungrammatical at times.

   If one is interested in only Machine Translation like output, one can hide all tother rows and see only the final layer output as shown in the figure 6.



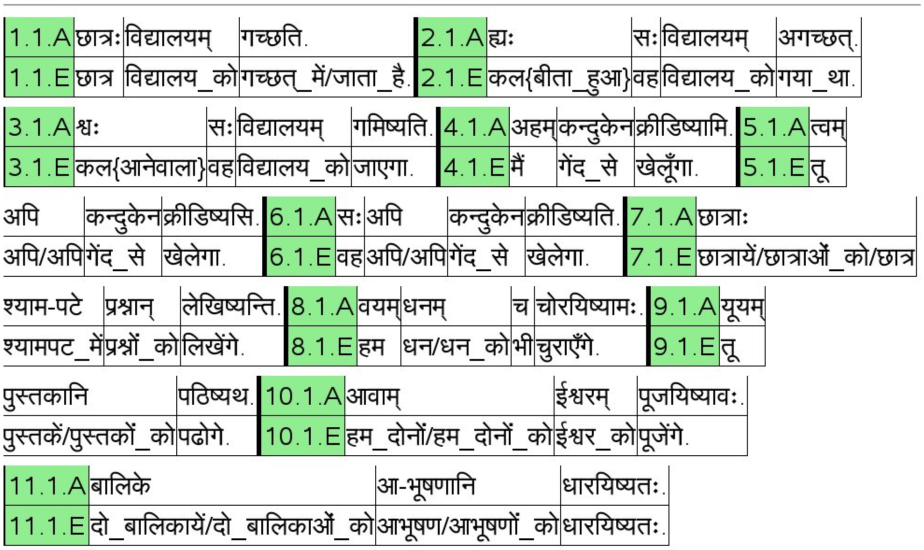
गच्छति.	2.1.A	ह्यः
गच्छत्{7}{नपुं.ए.}/1_गम्लृ{1 लट् अ_एक}.	2.1.C	ह्यः{अव्य् }
गच्छत्{पुं_ए.}+{में#पर}/जा+{ता_है#0_रहा_है}{अ_एक}.	2.1.D	कल{बीता_हुआ}
गच्छत्_में/जाता_है.	2.1.E	कल{बीता_हुआ}

सः	विद्यालयम्	अगच्छत्.	3.1.A	श्वः
तत्{1}{पुं.ए.}	विद्यालय{2}{पुं.ए.}	1_गम्लृ{1 लङ् अ_एक}.	3.1.C	श्वः{अव्य् }
वह{पु.}{पुं_ए.}+{0#ने}	विद्यालय{पुं_ए.}+{को2}	जा+{या_था}{अ_एक}.	3.1.D	कल{आनेवाला}
वह	विद्यालय_को	गया_था.	3.1.E	कल{आनेवाला}

सः	विद्यालयम्	गमिष्यति.	4.1.A	अहम्
तत्{1}{पुं.ए.}	विद्यालय{2}{पुं.ए.}	1_गम्लृ{1 लृट् अ_एक}.	4.1.C	अस्मद्{1}{ए.}
वह{पु.}{पुं_ए.}+{0#ने}	विद्यालय{पुं_ए.}+{को2}	जा+{गा}{अ_एक}.	4.1.D	मैं{पुं_ए.}+{0#ने}
वह	विद्यालय_को	जाएगा.	4.1.E	मैं

कन्दुकेन	क्रीडिष्यामि.	5.1.A	त्वम्	अपि
कन्दुक{3}{पुं.ए.}	1_क्रीड़ृ{1 लृट् उ_ए}.	5.1.C	युष्मद्{1}{ए.}	अपि{अव्य् }/अपि{अव्य् }
गेंद{पुं_ए.}+{से3}	खेल+{गा}{उ_ए}.	5.1.D	तू{पुं_ए.}+{0#ने}	अपि/अपि

Figure 5: Sanskrit text – with hindi generation



1.1.A	छात्रः	विद्यालयम्	गच्छति.	2.1.A	ह्यः	सः	विद्यालयम्	अगच्छत्.
1.1.E	छात्र	विद्यालय_को	गच्छत्_में/जाता_है.	2.1.E	कल{बीता_हुआ}	वह	विद्यालय_को	गया_था.

3.1.A	श्वः	सः	विद्यालयम्	गमिष्यति.	4.1.A	अहम्	कन्दुकेन	क्रीडिष्यामि.	5.1.A	त्वम्
3.1.E	कल{आनेवाला}	वह	विद्यालय_को	जाएगा.	4.1.E	मैं	गेंद_से	खेलूँगा.	5.1.E	तू

अपि	कन्दुकेन	क्रीडिष्यसि.	6.1.A	सः	अपि	कन्दुकेन	क्रीडिष्यति.	7.1.A	छात्राः
अपि/अपि	गेंद_से	खेलेगा.	6.1.E	वह	अपि/अपि	गेंद_से	खेलेगा.	7.1.E	छात्रायें/छात्राओं_को/छात्र

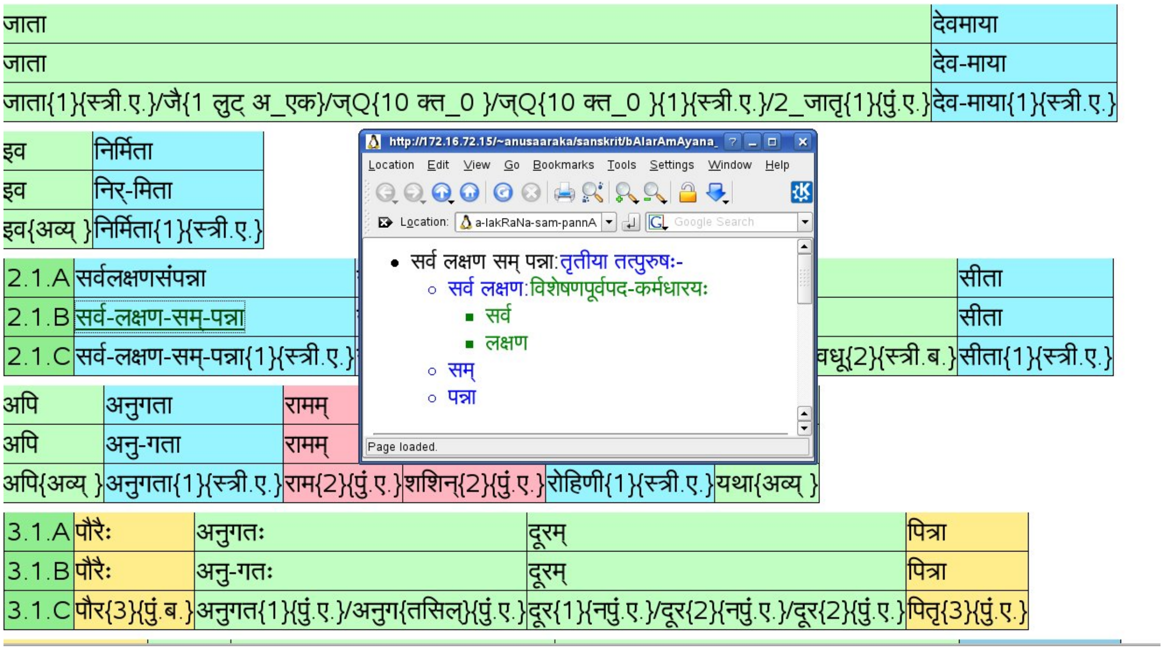
श्याम-पटे	प्रश्नान्	लेखिष्यन्ति.	8.1.A	वयम्	धनम्	च	चोरयिष्यामः.	9.1.A	यूयम्
श्यामपट_में	प्रश्नों_को	लिखेंगे.	8.1.E	हम	धन/धन_को	भी	चुराएँगे.	9.1.E	तू

पुस्तकानि	पठिष्यथ.	10.1.A	आवाम्	ईश्वरम्	पूजयिष्यावः.
पुस्तकें/पुस्तकों_को	पढोगे.	10.1.E	हम_दोनों/हम_दोनों_को	ईश्वर_को	पूजेंगे.

11.1.A	बालिके	आ-भूषणानि	धारयिष्यतः.
11.1.E	दो_बालिकायें/दो_बालिकाओं_को	आभूषण/आभूषणों_को	धारयिष्यतः.

Figure 6: Sanskrit text – with only hindi output

5. Sanskrit is very rich in samāsa formation, as well as its usage. We provide a hyper link to the analysis of samāsa, as shown in the fig 7. As is obvious from the interface, after adding necessary modules, this anusāraka also leads to a full fledged MT system.



जाता	देवमाया
जाता	देव-माया
जाता{1}{स्त्री.ए.}/जै{1 लुट् अ_एक}/ज्Q{10 क्त_0 }/ज्Q{10 क्त_0 }{1}{स्त्री.ए.}/2_जातृ{1}{पुं.ए.}	देव-माया{1}{स्त्री.ए.}

इव	निर्मिता
इव	निर्-मिता
इव{अव्य् }	निर्मिता{1}{स्त्री.ए.}

2.1.A	सर्वलक्षणसंपन्ना		सीता
2.1.B	सर्व-लक्षण-सम्-पन्ना		सीता
2.1.C	सर्व-लक्षण-सम्-पन्ना{1}{स्त्री.ए.}	वधू{2}{स्त्री.ब.}	सीता{1}{स्त्री.ए.}

अपि	अनुगता	रामम्			
अपि	अनु-गता	रामम्			
अपि{अव्य् }	अनुगता{1}{स्त्री.ए.}	राम{2}{पुं.ए.}	शशिन्{2}{पुं.ए.}	रोहिणी{1}{स्त्री.ए.}	यथा{अव्य् }

3.1.A	पौरैः	अनुगतः	दूरम्	पित्रा
3.1.B	पौरैः	अनु-गतः	दूरम्	पित्रा
3.1.C	पौर{3}{पुं.ब.}	अनुगत{1}{पुं.ए.}/अनुग{तसिल्}{पुं.ए.}	दूर{1}{नपुं.ए.}/दूर{2}{नपुं.ए.}/दूर{2}{पुं.ए.}	पितृ{3}{पुं.ए.}

http://172.16.72.15/~anusaaraka/sanskrit/bAlarAmAyana_
Location Edit View Go Bookmarks Tools Settings Window Help
Location: a-lakRaNa-sam-pannA
Google Search
Page loaded.

- सर्व लक्षण सम् पन्ना:तृतीया तत्पुरुषः-
    - सर्व लक्षण:विशेषणपूर्वपद-कर्मधारयः
        - सर्व
        - लक्षण
    - सम्
    - पन्ना

Figure 7: samkshipta rāmāyana

In order to develop such a anusāraka, one needs several modules such as morphological analyser, sandhi splitter, samāsa handler, pos tagger, parser, word sense disambiguator and finally a target language generator.

Akshar Bharati group has developed some of these modules(Bharati,2006). They are available at http://sanskrit.uohyd.ernet.in. In this session, we have two presentations related to the morphological analysis and generation of Sanskrit, therefore I'll make this floor open to our two invitees, by indicating the complexity of the task involved.

3 Computational Tools for Sanskrit

It is believed that compared to the task of applying Paninian Grammar Formalism to other languages, the task of developing computational tools for Sanskrit is much easier in view of existence of ashtādhyāyi. It is really perplexing that in spite of all available resources, still one finds it difficult to various computational tools for analysis of Sanskrit. One of the reasons is that the whole literature is still inaccessible to the computer scientists and the Sanskrit scholars rarely turn towards computer science.

The complexity of word formation in Sanskrit may be illustrated by the finite state automata in fig 8. The '*' indicates the starting node of the automata.

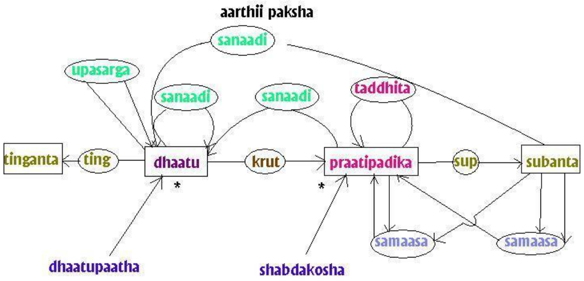


Figure 8: Word Formation in Sanskrit

Thus both the 'pratipadika's as well as 'dhatu's provide the starting point. The first level of conjugation involves only two pratyayas – viz. sup and tiṅ.
Thus we have
pratipadika + sup - > subanta e.g. rāmeṇa
dhatu + tiṅ - > tiṅanta e.g. gacchati

There are few kridanta suffixes, which produce 'avyaya's.
e.g. ktvā, tumun, etc. as in
gam + ktvā - > ; gatvā
gam + tumun - > ; gantum

Some of the kridanta suffixes produce new pratipadikas, and thus they take additionally one more suffix viz. sup, to produce a subanta, as in
gam + śatṙ -> gacchat. This further takes optionally a feminine suffix which is then followed by a sup. So one may have a form such as 'gacchati' which may be analysed as
gam + śatṙ + sup
This results in the second level of word formation requiring two suffixes viz. kṛt and sup.
Other paths that produce new pratipadikas or dhatu's are

- pratipadika + sanādi suffix
  ex: putra + kyac -> putrīyati
  putra + kāmyac -> putrakāmyati

kṙṣṇa + kvip -> kṙṣṇati

- dhātu + sanādi suffix
  ex: pipaTaṣti / bobhūyate / gopayati, etc.

- upasarga + dhātu ->
  ex. pra + hṙ / A + hṙ / etc.

- The taddhita suffixes generate new pratipadikas, as in daśaratha to dāśarathi.

New pratipadikas may also result from compound formation. There are 6 ways of compound formation, viz:
sup + sup
sup + tiṅ
sup + pratipadika
sup + dhātu
tiṅ + sup
tiṅ + tiṅ

Though there are 6 possibilities, only some of them are very productive, and others are very rare. It was found that around 20-25% of the words in Sanskrit text are compounds. The complexity is further aggravated by the extensive sandhi formation in Sanskrit.The mandatory sandhi in the formation of compounds makes it a kind of deadlock situation as illustrated in fig 9.

Thus there is a kind of deadlock situation. However, practically one can break this deadlock by developing a morphological analyser that handles first level suffix, viz. tiṅ and sup. It is found that almost 50 to 60% of the words are analysed at this layer. A separate sandhi splitter which takes inputs from this morphological analyser can then be developed independently which can then handle the samāsas also.

Now I invite Dr. Girish Nath Jha followed by Dr. Malhar Kulkarni to make their presentations.

References

[1] Bharati Akshar, Vineet Chaitanya, Rajeev Sangal, *NLP A Paninian Perspective,* Prentice Hall of India, Delhi,1994

[2] Kulkarni, Amba P., *Design and Architecture of anusAraka: An Approach to Machine Translation,* Satyam Techical Review vol 3, Oct 2003, pp 57-64

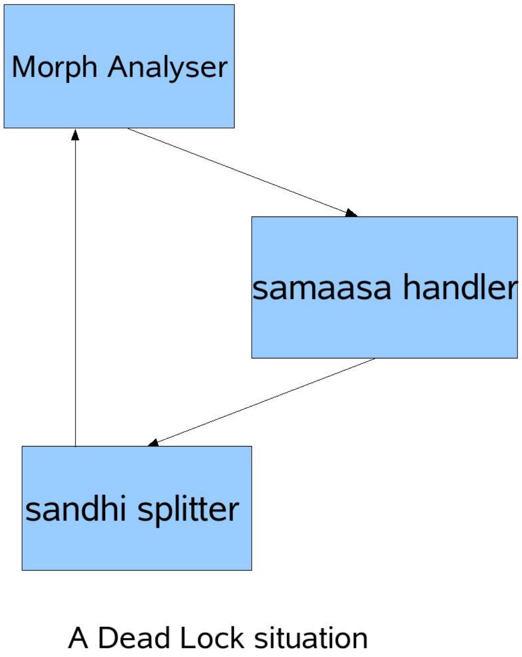


Figure 9: Deadlock in word analysis

[3] Bharati, Akshar, Amba P Kulkarni, *English from Hindi viewpoint: A Paninian perspective,* Platinum Jubilee conference of LSI at HCU, Hyderabad, Dec 6-8, 2005

[4] Bharati, Akshar, Amba P Kulkarni, V Sheeba, *Building a Wide Coverage Morphological Analyser for Sanskrit: A Practical Approach,* invited speech at 'First National Symposium on Modeling and Shallow Parsing of Indian Languages', 31st March - 4th April 2006, IIT Mumbai