Developing a Sanskrit Analysis System for Machine Translation

Girish N. Jha, Sudhir K. Mishra, R. Chandrashekar,
Priti Bhowmik, Subash, Sachin Mendiratta, Muktanand

Special Center for Sanskrit Studies,
Jawaharlal Nehru University, New Delhi-110067
girishj@mail.jnu.ac.in

The authors in the present paper are outlining the parameters of a Sanskrit Analysis System (SAS). Importance of this system can be understood by the fact that no translation system from Sanskrit to Indian languages can be developed without first building such systems. The paper gives details of each component required, and current developments. In particular, the paper will focus on the following components –

- Building linguistic resources for translation
- Reverse *Sandhi* module for initial segmentation
- POS tagging module
- Verb inflection morphology (*tiṅanta*) analysis module
- Nominal Inflection morphology (*subanta*) analysis module
- Derivational morphology (*kṛt*, *taddhita*, *strī*, *samāsa*) analysis module
- *Kāraka* analysis module

Building robust and quality Machine Translation Systems (MTS) has been one of the most challenging endeavors for the Artificial Intelligence (AI) and Computational Linguistics (COLING) community so far. The Source Language (SL) and Target language (TL) dynamics and the direction of translation are difficult variables in determining the complexity level of the translation system. Building a MTS with Sanskrit as SL is important, interesting as well as challenging. It is important because Sanskrit is the only language in India which can be truly considered a 'donor' language. The vast knowledge reserves in Sanskrit can be 'transferred' (read translated) to other Indian languages with the help of computer. For various historical reasons, this knowledge in its same un-diluted form may not have been allowed to travel to regional languages/cultures. It is interesting because Sanskrit has a unique status in various senses, one of them being the fact that it is a language 'frozen' in time. It is a highly standardized language with little scope for deviation from the system of Pāṇini, and is the only national language of India with no state in India. Having Sanskrit as the SL is challenging because of the difficulty in parsing due to its synthetic nature in which a single word (sentence) can run up to 32 pages (Baṇbhatta's Kādambarī).

1. Building linguistic resources for translation

Building/collecting/adapting suitable corpora are essential for any successful MTS. The advantage with Sanskrit is that there are many online texts available. We will be mentioning a few later. The problem with most of them is the following –

- they are mostly of Sanskrit kāvya texts in metric compositions

- are written as PDF in weird custom fonts thereby making any conversion very difficult
- there are very few digital dictionaries available and not all of them can be adapted as critical resource

1.1 Adapting Monier Williams Digital Dictionary (MWDD) by Louis Bontes for tagging and other processing

MWDD created by Louis Bontes is a good computational resource for Sanskrit due to the fact that nearly 200,000 words are stored in a text file downloadable from the net. However, this dictionary needs to be adapted depending on what we want to use it for. For example, if we want to use it for reverse sandhi application, then we have to just use the basic word list file (after having converted the Harvard Kyoto transliteration scheme to a ITrans) and then create a reverse dictionary programmatically for checking the words right to left. However, if we are going to use it for subanta recognition and analysis, then we will have to store additional information like category and gender for the nominal bases (prātipadikas). If we are going to use it for identifying the verbs (or for Kāraka analysis) then it might be a better idea to store verbs and their forms separately

Currently, work is underway by MA students of this center to take the basic word list from MWDD, convert it into ITrans, add the basic information like category, gender, number etc for all the entries using ';' as the separator. The java engine will read this file and temporarily 'remember' the information pertaining to an entry for processing.

1.2 e-Corpora hunt online or elsewhere

Collecting available e-corpora is an arduous task as well. There are many etexts available online, but many of them are un-usable due to following reasons - they are written as pdf files with weird fonts, they are mostly non-prose, non-contemporary literature. There are others which can be used with some modifications like converting the transliteration scheme to ITrans and performing some normalization on them. Tirupati Sanskrit Vidyapeetham is reportedly building a sandhi-free corpus. We can use it as well.

1.3 Building custom corpora

Under this endeavor we are developing a basic sentence list from Apte's book on Sanskrit comprehension. Also we are building a basic dhātupāṭha list. Of the main entries in this list we have identified 500 commonly occurring verb roots whose conjugational forms need to be stored. Of these 500 forms, some verb roots will be marked for additional conjugations as reduplications etc.

1.4 Building *amarakoṣa*

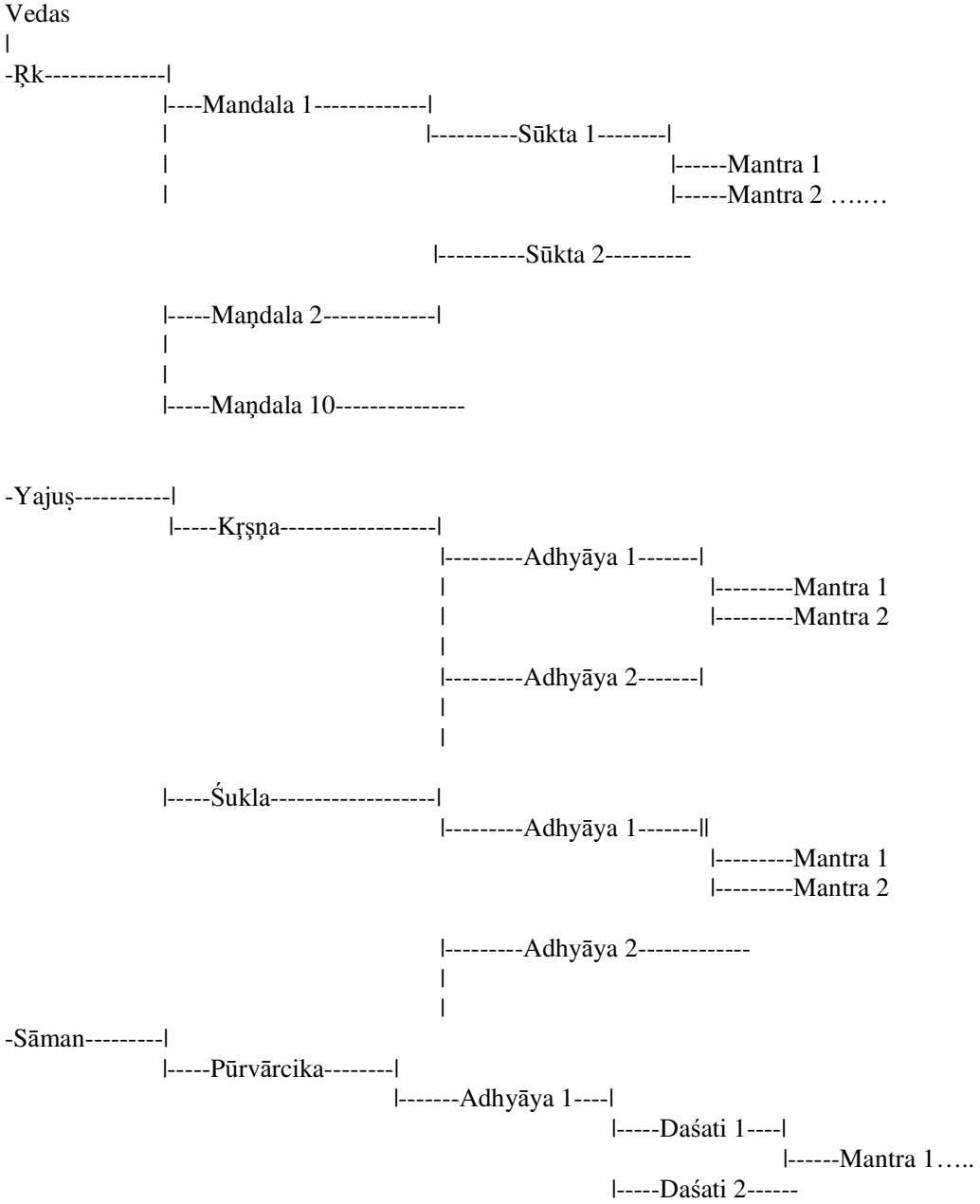
A basic structure database with Java front end has been developed. This lets us store up to 50 synonyms of each base entry with multilingual (up to 3 at this point) equivalents.

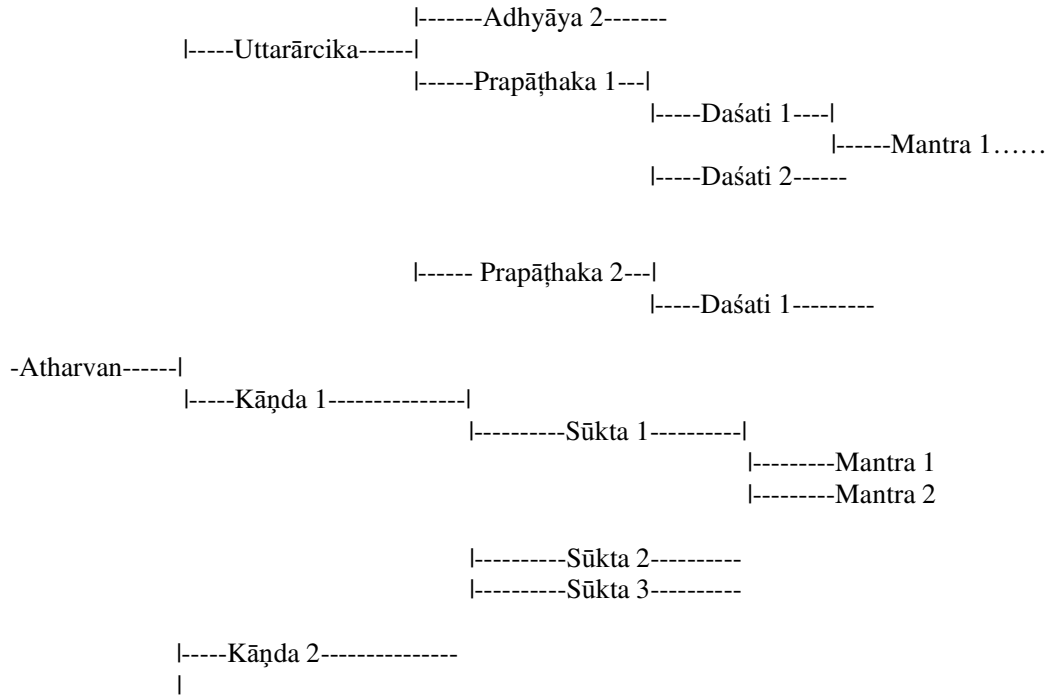
The system has been developed as an online system, therefore data entry should not be a problem.

1.5 Vedic database

The work has started as an online system with add/edit and search data facility for the three Vedas. This work aims at developing a relational database system as an e-corpus for the storage and interactive access of the Vedic texts. Using technologies such as SQL Server, an RDBMS based system and Java Server Pages, the system will work as an interactive and multi-dimensional knowledge base system for Vedas.

The database design is given as –





The related tables in the back-end, i.e. database will be as follows :

ṚgVedaSūkta

id	Sūkta_devata	Ṛṣi	Sukta descry.
1	Agni	maducCandA	triShTup CandaH
2	vAyu		

ṚgVedaṚcā

Id	Sūkta id	Ṛcā_id
1	1	1.1
10	2	2.1

ṚgVedaṚcāDetails

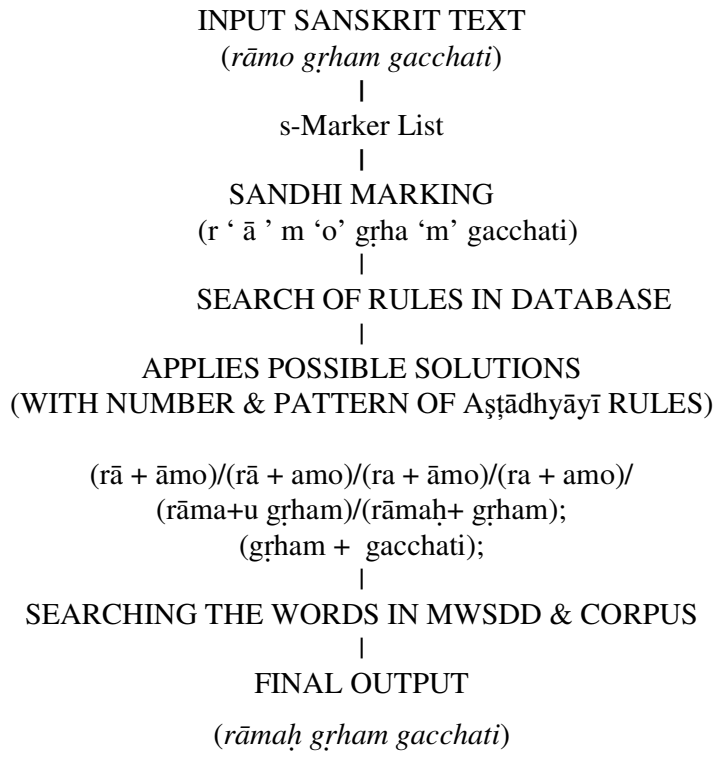
Sūkta_ṛcā_id	devatā	Ṛcā _text	Eng_trans	Hindi_trans
1	Agni	agnimILe purohitaM…….	I Laud Agni, the chosen Priest,…..	MaiM purohita , yajna ke…….
10	vAyu	vAyavA yAhi darSateme somA	Beautiful Vayu, come, …..	He darSanIya vAyu !....

2. Reverse Sandhi module for initial segmentation

A *Sandhi* analyzer based on Paninian formalism is being developed with the help of AD rules and MWDD adaptations. While some attempts have been made (Huet, 2005) to develop string segmentation engines based on ad-hoc processing and lexical search, there is no *Sandhi* analyzer which comprehensively analyzes a Sanskrit text according to Pāṇinian formalism. This module will be the first step towards a Sanskrit analysis system. The work will also be helpful for self-reading and understanding of Sanskrit texts by those readers who do not know or want to go through the rigors of *Sandhi viccheda*. It will also be helpful for Sanskrit –interpretation and summarization of text. The objectives of this Research & Development (R&D) are the following -

- to build a rule-base of Pāṇinian *sandhi* rules for *Sandhi* identification and analysis
- to adapt Monier Williams Sanskrit Digital Dictionary (MWSDD) for *Sandhi* analysis purposes
- to adapt available e-corpora and customize them and for *Sandhi* analysis purposes

The design of the proposed *Sandhi-Viccheda* system is as follows-



3. POS tagging module

For a given *sandhi*-free Sanskrit text and also transcribed Sanskrit speech the proposed system will assign correct POS tag for each word.

The research aims at two main objectives.

- evolve a tag-set for Sanskrit text and speech with tags having Sanskrit acronyms.
- build a POS tagger for Sanskrit.

This POS program can be used in several Natural Language Processing (NLP) related applications for Sanskrit language like:

- MT
- Speech recognition/synthesis
- Information retrieval /data mining
- Word-sense disambiguation
- Corpus analysis of language & lexicography

There are numerous POS taggers available for English and other European languages like Brill's POS tagger, Claw's POS tagger etc. But for Sanskrit and Indian Languages there is not much work done. A theoretical model for POS tagger for Sanskrit has been evolved by Gerard Huet (2003), with a partial application in his website. A Morphological Analyser for Sanskrit has been made by IIIT group in collaboration with Melkote Sanskrit Academy. Rashtriya Sanskrit Vidyapeeth, Tirupati is creating a Corpus for Sanskrit which is Sandhi-free. Apart from these many researches have been made and many in process in major Indian Languages. Some of them are listed below -

- Hindi tag-set by Patrick Hall
- Tamil Morphological tagger, Vasu Renganathan
- Hindi POS tagger, Sushama Bendre, Central University of Hyderabad

A sample tagging example is given below –

Sample POS tags for Sanskrit Verbs

P – parasmai pada
A – Atmane pada
K – karmaNi
N – Nijanta
S – sannanta
laT – laT lakaara
liT – liT lakaara
luT – luT lakaara
lR^iT – lR^iTlakaara
la^N – la^N lakaara
vli^N – vidhili^N lakaara
aali^N –aashiirli^n lakaara
lu^N – lu^N lakaara
lR^iN – lR^iN lakaara
loT – loTlakaara
1.1 –prathamapuruSha ekavachana
1.2 –prathamapuruSha dvivachana

1.3 –prathamapuruSha bahuvachana
N – naamapada
NV – naamavisheShaNa
SN – sarvanaamapada
p – puMli^Nga
s – striili^Nga
n – napuMsakali^Nga
t – trili^Nga
Av – avyaya
Avk - kaala-vaachi avyaya
Avd – desha-vaachi avyaya
KVAv – kriyaavisheShaNa avyaya
tAv – tumunnantaavyaya
LAv - lyabantaavyaya

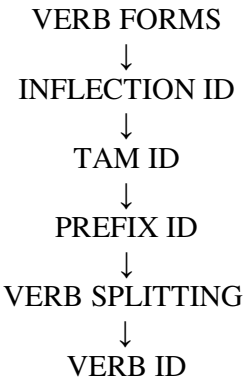
Manual tagging of a sample Sanskrit text using the sample tagset

dashakumaaracharitam
vishrutacharitam naama aShTamaH uchChvaasaH
(MLBD, page 210-212)

vi-achintayaM_**Pla^N3.1** cha_**Av** sarvaH_**SNp1.1** api_**Av** atishuuraH_**NVp1.1**
sevakavargaH_**Np1.1** mayi_**SNt7.1** tathaa_**Av** anuraktaH_**NVK2a1.1** yathaa_**Av**
aaj~nayaa_**Ns3.1** jiivitam_**Nn2.1** api_**Av** tR^iNaaya_**Nn4.1** manyate_**AlaT1.1**
raajyadvitayasainyasaamagryaa_**Ns3.1** cha_**Av** na_**Av** aham_**SNt1.1**
ashmakeshaat_**Np5.1** vasantabhaanoH_**Np5.1** nyuunaH_**NVp1.1**
niityaaviShTaH_**NVK2ap1.1** cha_**Av** ataH_**Av** vasantabhaanuM_**Np2.1**

4. Verb inflection morphology (*tiṅanta*) analysis module

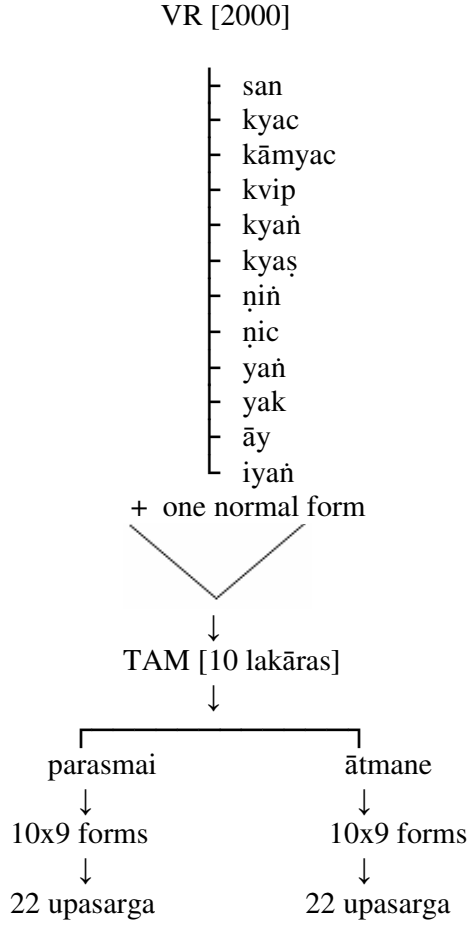
This module will correctly describe verbs in a *laukika* Sanskrit text. Verbs play an important role in syntactico-semantic relations in the sentence. The overall idea is to identify and analyze the Sanskrit verbs correctly so that any Sanskrit to Indian language machine translation can benefit from this processing. The overall model of the system is as follows-



Sanskrit verb forms are very complex. They carry tense, aspect, person, number information all in the inflection forms. Besides, they can also contain derivations containing semantic informations like causation, desire, repitition, negation etc. Therefore it becomes very difficult to split out the verb and separate the verb root and complex information units encoded in it. Sanskrit has about 2000 verb roots classified in 10 morphological and semantic classes called gaṇas, and can also be further sub-classified as

- normal forms
  - without any of the 12 derivational affixes
    - 11 listed by Pāṇini [P 3.1.32]
    - 1 more 'kvip' added by Kātyāyana
- derived forms
  - ṇijanta (causative – ṇic)
  - sannata (expressing desire – san, kyac, kāmyac, kvip, kyaṅ, kyaṣ,ṇiṅ, yak, āy and iyaṅ)
  - yaṅanta (duplicated – yaṅ and yaṅluṅant)

Then these can have ātmane and parasmai forms in 10 lakāras and 3 x 3 person and number combinations. Then these can also be potentially prefixed with 22 prefixes. Finally there could be in-numerable nāmadhātus (nominalized verbs).



Therefore the approach followed by many to store all the Sanskrit verb forms is not going to work. Therefore we propose storing the commonly used verb forms (around 640 verb roots) and reverse Pāṇinian approach to parse out other verb forms not stored. The processing sequence will be as follows -

- the verb inflection (*parasmai* / *ātmane*) is identified from database and a rough guess is made about the verb,
- the *lakāra* information based on inflection is obtained,
- evaluate each of the 12 derivational affixes
- identify if there are any of the 22 prefixes
- the verb root is determined by weeding out other elements and database matching.

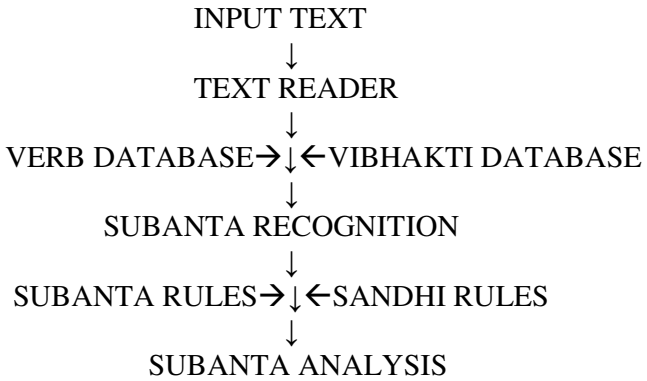
The verb root thus identified with all the other potential components grammatical information gathered in the form of verb tags can be potentially used in a machine translation system performing translation from Sanskrit.

5. Nominal Inflection morphology (*subanta*) analysis module

A Sanskrit sentence has all padas (except the verb) as subanta padas. Therefore it will be essential to first identify those and then analyze those into their morphological constituents for the next step of processing. The proposed module has the following components -

- a V*ibhakti* database of Subanta morphemes and allomorphs for Subanta recognition in a sentence
- a database of Subanta and Sandhi rules required for morphological analysis

The overall model of the subanta analyzer is as follows-



Sample Illustration

Step 1→ rāmlakṣmanau sundaram nagaram paśyatah.

Step 2→ Recongnition of verb- *(paśyati)*

Step 3→ Recognition of *subantas* (*Rāmlakṣmanau sundaram nagaram*)

Step 4→ Analysis of *Subantas*

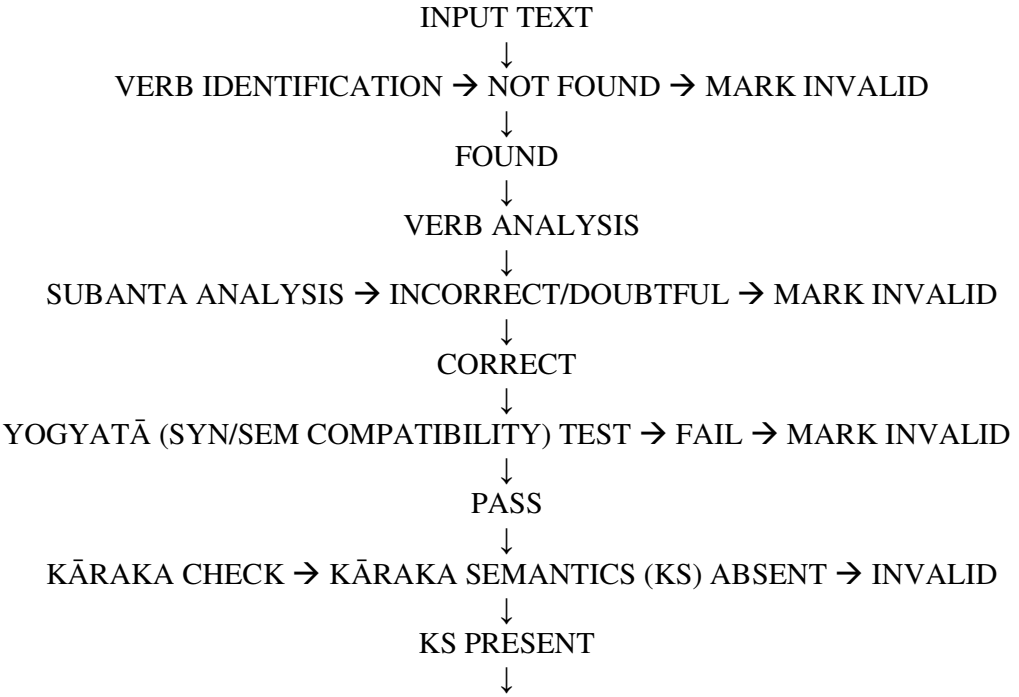
Step 5→ rāmalakṣmanau [PRATHAMAA_DVIVACHANA] → raamalakshmana [PRAATIPADIKA] + au + [SUP_ PRATH_DVI] sundaram[DVITĪYĀ _EKAVACHANA] → sundara [PRĀTIPADIKA] + am + [SUP_DVIT_DVI) nagaram [DVITĪYĀ_EKAVACHANA] → nagara [PRĀTIPADIKA]+ am [SUP_DVIT_EKA]

6. Derivational Morphology (krdanta/taddhita) and compounds (samasa) analysis module

After the nominal base (prātipadika) has been isolated from the nominal inflection (sup), then a dictionary check will be performed for possible translation into the target language. If the word is not found in the dictionary, then it is assumed to be a complex form which can be further broken down into its derivational constituents (krdantas, taddhitas, samasas and feminine forms (strī)).

7. Karaka analysis module

This module will provide the Karaka analysis for the input Sanskrit text. The overall model of the system is as follows-



KĀRAKA ANALYSIS
→ KĀRAKA
→ DEF SUTRA
→ EXPLANATION
→VIBHAKTI
→ DEF SUTRA
→ EXPLANATION
↓
OUTPUT TEXT

This module has the database representation of essential and optional syntactico-semantic parameters, exceptions, parallel applications for all the six kārakas of Sanskrit. Pāṇini describes two more constructions as vibhaktis - ṣaṣṭhī (genitive) and saṃbodhana (vocative). Table 1 lists basic defining principles of each kāraka and total number of rules describing the entire range of syntactico-semantics -

k_id	k_name	definition	rule_base
1	Karman	1.4.49	1

The following table provides the rule details by unifying the field 'definition' from previous table to the field 'rule_id' of following table –

Rule_id	R_text	r_meaning
1.4.49	karturīpsitatamaṃ karma	The one which is most desired by the agent

The next table (by obtaining the karaka id from the first table) provides the ordered listing of syntactico-semantic information pertaining to each kāraka rule application, and also lists exceptions or parallel applications if any -

K_i d	Ss1	ss2	e1	p1
1	{(Agt)Nom (N_Acc)K2}S_Act	{(Agt)Instr N_Nom)K2}S_Pass	1	0

Here, a positive integer value in the e1 field means that there are exceptions which can be found by unifying this field value with the corresponding table. A value of integer 0 in the p1 field means that there are no parallel applications of this rule.

The next table lists the indeclinables which are generally exceptions to most kāraka rules. For example the field e1 from the second table will be unified with field eid in the third table to obtain exceptions -

eid	Words	vibhakti	kāraka
1	antarā	2	0
1	Antareṇa	2	0

8. Conclusion

The SAS components presented in the paper are still in the research and development phase and are likely to finish up in about two years time. The actual application of this system for translating Sanskrit to Indian languages will begin thereafter. The work is being carried out by the M. Phil. and Ph.D. students of the Sanskrit Center of JNU, and this paper aims to get feedback from the seminar and readers of the paper so that the system design can be tuned up and potential errors can be avoided.

REFERENCES

BHARATI, A., Sangal R., 1990, A karaka based approach to parsing of Indian languages, *proc of the 13th COLING vol 3, pp 30-35*, Finland.

BHARATI, Akshar, Chaitanya, Vineet, Kulkarni, Amba P., Sangal, Rajeev, LTRC, IIIT, Hyderabad, http://www.iiit.net/ltrc/Publications/Techreports/tr010/anu00kbcs.txt (accessed: 7 April 2005).

BHARATI, Akshar, Vineet Chaitanya and Rajeev Sangal. , Mar.-Dec. 1991a, *A Computational Grammar for Indian languages processing*, Indian Linguistics Journal, 52, 91-103.

BRIGGS, Rick, 1985, Knowledge representation in Sanskrit, *AI magazine*.
CELEXTEL'S Online Spiritual Library, http://www.celextel.org , (accessed on Aug 10, 2005)

FILLMORE, Charles J. 1968, *The case for case*, In E. Bach and R.T. Harms (eds.), Universals of Linguistic Theory, Holt Rinehart and Winston, New York, pp. 1-88.

HUET, G'erard, 2003, Towards Computational Processing of Sanskrit, Recent Advances in Natural Language Processing : Proceedings of the International Conference ICON-2003, Mysore, India.

HUET, Gerard, http://pauillac.inria.fr/~huet/SKT/sanskrit.html (accessed: 7 April 2005).

JHA, Girish N, MISHRA, Sudhir K, R. Chandrashekar, 2005, Information Technology applications for Sanskrit lexicography: the case of Amarakosha, Proc. of *ASIALEX 2005,* Singapore, pp. 104-08.

JHA, Girish N., 1993, *Morphology of Sanskrit Case Affixes: A computational analysis*, M.Phil dissertation submitted to J.N.U., New Delhi.

MISHRA, Sudhir K, Jha, Girish N, 2004, Karaka Analyzer for Sanskrit, In proc. of *SALA-24*, New York.

MISHRA, Sudhir K, Jha, Girish N, 2004, Sanskrit Karaka Analyzer for Machine Translation, In *SPLASH proc. of iSTRANS,* Tata McGraw-Hill, New Delhi, pp. 224-225.

MISHRA, Sudhir K, Jha, Girish N, 2005, Identifying Verb Inflections in Sanskrit Morphology, In *proc. Of SIMPLE 05, pp 79-81*, IIT Kharagpur .

RCILTS website, JNU, http://rcilts.jnu.ac.in , (accessed on July 10, 2005)
RC-ILTS-Oriya, Utkal University, http://www.ilts-utkal.org/oriyagrammar.htm (accessed: 7 April 2005).

SANGAL, Rajeev, 'Shakti Machine Translation System' IIIT, Hyderabad. http://www.elitexindia.com/paper2004/rajeevsangal.doc (accessed: 7 April 2005).

SANSKRIT E-TEXTS, http://www.sacred-texts.com , (accessed on July 15, 2005)

SHARMA, R.N., 1995, *The Aṣṭādhyāyī of Pāṇini* : Munshiram Manoharlal, Delhi.

SINHA, R.M.K, 1989, "A Sanskrit based Word-expert model for machine translation among Indian languages", *Proc. of workshop on Computer Processing of Asian Languages*, Asian Institute of Technology, Bangkok, Thailand, Sept.26-28, 1989, pp. 82-91.12.

TDIL brief on *śābdabodha*, Gov. of India, http://tdil.mit.gov.in/download/Shabdabodha.htm (accessed: 7 April 2005).

TDIL brief on *deśikā*, Gov. of India, http://tdil.mit.gov.in/download/desika.htm (accessed: 7 April 2005).

VIVEK Quarterly journal on Artificial Intelligennce, NCST, Mumbai, http://www.ncst.ernet.in/kbcs/vivek/issues/13.2/sumam/sumam.html (accessed: 7 April 2005).