SANSKRIT AND COMPUTER

(Proceedings of the U.G.C. National Seminar)

Edited by

Dr. Keshab Chandra Dash, D.Litt.
Reader & Head,
Department of Nyāya Philosophy
Sri Jagannath Sanskrit Vishvavidyalaya,
Puri

1995
PRATIBHA PRAKASHAN
DELHI

must end. It is being severelly challenged as the most favoured conception of anglosaxions later supported by the logical empiricists many of whom became American. The most powerful alternative framework has been offered by the Hermenautists who are unfortunately so much divided among themsleves, one is afraid to call oneself a Hermenautist lest people take him to be one who wants to be excused for all his wooly thinking. I for one do not wish to associate myself with this group. But this view of Hermenautics is largely a creation of those who would make science, even as it is narrowly conceived by the logical empirisity, the whole of human knowledge a tradition with which even Patnam wanted to maintain distance. But philosophical hermenautic shows a way to different notion of knowledge and truth. It brings closer theory and practice and cures us of the idolatry of science which if it is the modern European science interpreted as narrowly as the logical empiricists must be got rid of . On the other hand, closely understood reference to text and interpretation involved in hermenautic project to any mind comes closer to the project of Prasthāna Mīmāṃsā which is contrary to craze for individual's originality satisfied often by free lance thinking which is not only not rooted to one's culture but often repeats in much inferior style and diction what people in the tradition have already written. Entire English Philosophy of the modern period may not be have foot note, but even if it is, that foot note must be identified as footnote and then real progress would be possible. This is what Professor Bhattacharya insisted on in the Indian context. *Anucchṛṅkhalacintā* or *Prasthāna Mīmāṃsā* rather than *uchṛṅkhalacintā Pārāyaṇa* is not nor had ever been in India the model of philosophizing.*

* Address of the chief Guest.

ISSUES IN COMPUTER-BASED SANSKRIT STUDIES

P. Ramanujan

(धीरो मनीषी ज्ञः प्राज्ञः संख्यावान् पण्डितः कविः - अमरकोशः)

ABSTRACT

In this paper, we discuss some of the issues involved in using Computers and Sanskrit in tandem. The desirability of using Computational (Natural Language Processing - NLP) approaches to Sanskrit teaching and research and seeking clues to solve riddles in Language Processing from the wonderful grammar rules of Sanskrit by sage Pāṇini is highlighted. Experience from our efforts in DESIKA project is provided to supplement the points made.

We also try to the lid off the topic of Sanskrit language benefitting Computer Science field through the role of a dedicated Computer programming language, as such a notion appears to exist in the minds of quite a good number of followers of this inter-disciplinary area of research. A few ideas are also fancied towards this end.

An awareness regarding the amount of success achieved at C-DAC in this area is also intended to be created together with our thinking for future work on the topic.

INTRODUCTION

In the field of Artificial Intelligence in Computer Science, efforts are on to devise Computers that can be useful in various day-to-day activities of human beings. Most of these acts involve some sort of communication through 'natural (spoken) languages. These natural languages have many features shared between the communities speaking those languages. For Computers to be useful to man in such situations, we need to computationally handle these 'natural' language.

The first task in such an exercise is to reduce the grammar of the concerned language to a 'Computable' form and then attempt its processing in the Computer. Here, the issue of

paramount importance is not the expressivity of a natural language, but the exhaustive coverage of all the usable word-forms by its grammar and the amenability of its grammar to algorithmisation for mechanical processing. If the spoken form of the language is also covered well by the grammar, it will be an added advantage.

We also require a conceptual classification of objects denoted by natural languages (i.e. word-meanings) in an elegant and proper manner. This would help determine the meanings of natural language utterances. There is also a need for clear rules and methods to disambiguate word-meanings when multiple senses are admissible to words.

A diverse range of application areas with their own peculiar vocabulary and linguistic processes would also be required as a satisfactory test-bed for ascertaining the efficacy of the natural language processing formalism developed. Besides, it will be a real bonus if well-documented literature recording the results of systematic, comprehensive and time-tested research endeavours is available. An integrated system incorporating such ambitious requirements, if readily available, may rank on top of the possible methodologies to be chosen for the problem on hand, i.e. NLP.

As Sanskrit language possesses all these desirable features, it can be considered for systematic study to help in NLP issues in the Indian context. Majority of Indian languages are derived from or influenced by Sasnskrit, linguistically. Hence, work done on Sanskrit language would benefit the rest as well. Prominent areas to benefit would be Automatic translation between Indian languages, academic activities and literary pursuits (even for Non-Resident Indians).

On the other hand, if we consider the current status of Sanskrit language in the country, we find that it is not a popular dialect in conversational use, its scholars are few and far between and a general sense of diffidence and helplessness/fatalism prevails in them. This is despite the fact that they all are convinced and capable of demonstrating its virtues in traditional method. It is also a way of life and a noble duty for them to preserve and propagate the rich cultural heritage encoded in its vast literature starting with the Vedic Scriptures.

The modern education system has also contributed to the changed priorities of the current and future generations and the disregard for anything indigenous or homegrown if it is not in tune with or in demand in the western world. These very factors which look like dark clouds, provide the silver lining in the current discussion of a synthesis between Computing technology and Sanskrit language. As the synthesis will be effective only if both sides are on equal footing and contribute mutually on par, we wish to examine this issue and bring out the facts.

The Computer medium has emerged as a powerful tool in modern scientific investigations with more and more exact and other sciences being studied with their help. Thus, as a tool, if we use Computers to study, understand, preserve and propagate Sanskrit literature, it would enrich our linguistic knowledge and also find better acceptability in today's India. This is a challenging task, but certainly not insurmountable. This would essentially mean that the traditional Sanskritists may have to take to this medium and tool in a friendly manner (HINDI) and explore ways of adopting human-oriented methods to mechanical situations.

NLP researchers also could look up to the Sanskritists with due respects to their mastery of abstract subjects like Grammar, Phonetics, Epistemology, Metaphysics etc. besides Vedic texts, in order to seek appropriate solutions to conversational language processing. That this process has already started, albeit in a small way, is a happy augury for the future.

INITIATIVES

We now move on to the topic of the current scenario in this field, tracing a bit of recent history. From the past decade. Personal Computers are really becoming commonplace and serious effect of any activity at academic or industrial spheres inland or abroad have begun to be realised by larger number of common people. This has generated higher potential and demand for NLP products alongwith a better awareness on the subject.

To begin with, efforts at Dept. of Comp. Science &Engg., IIT, Kanpur, from 1983 onwards were directed at Indian language processing attempting to utilise the Sanskrit grammar

structure and particularly, Pāṇini's Kāraka Theory, for an interlingua-based machine translation system among pairs of Indian languages. This project has achieved considerable success by now, covering Hindi, Telugu, Kannada etc. as source or target languages for translation. Development of lexicons and conceptual structure design is also underway. Sanskrit is hidden as the root in this system.

The next major event in this field was the publication of two reports on the ideal nature of Shastraic (technical) Sanskrit for machine translation interlingua and the similarity of Vaiyakaṛana Shabda-bodha and Semantic network scheme of knowledge representation in AI, by Mr. Rick brings, a Scientist of NASA, USA, in AI Magazine, in 1984 and 85. The tone and tenor of these papers suggested that the conclusion has been drawn after trying quite a good number of other 'natural' languages as candidates for the purpose of a universal machine translation interlingua. This triggered a series of efforts both at home and abroad, in the field of Sanskrit and Computers, almost making the subject of topical interest in the Computer circles, then. Since then, numerous workshops, symposia, conferences etc. have been conducted in various parts of the country addressing different aspects. Further, the language of *Navya-ṇyaya* also has been found to be highly suitable for inferencing etc.

Through the UNDP- assisted Knowledge-Based Computer Systems (KBCS) project, work began at C-DAC, Pune, in 1989 to render the Pāṇini's grammar computationally. Further, through the Technololgy Development for India Languages (TDIL) project of Department of Electronics, Govt. of India, Sanskrit Institutions at Varanasi, New Delhi, Tirupati, Hardwar and Melkote are pursuing work in the field currently. Many other private and public Sanskrit organisations are also engaged in research and academic activities on the subject. The Graphics and intelligence based Script Technology (GIST) enables the use of and transilteration between Indian scripts in Computers.

SANSKRIT THRU' COMPUTERS

DATA ENTRY

One can now use Sanskrit language in Computers either for Word Processing needs (like a typewriter) or for

programming needs (to develop application software). For this, the necessary pre-requisite is the availability of a hardware or software solution to render the Devanagari (or other Indian script) characters on the Computer screen. For instance, among hardware solutions, installation of GIST PC Add-on card (of C-DAC), together with its firmware, in the Computer. The Indian scripts required and the keyboard stickers for them come alongwith the GIST kit. There are also many software solutions available for Devanagari (and other Indian script) usage in Computers, including ISM from GIST Group of C-DAC.

With any of these available, one can proceed with keying-in the data one wants, say, a particular word, text, poem etc. This constitutes data entry, requiring minimum knowledge regarding the basic Computer operations like booting, handling files etc. One should, of course, check for all Sanskrit characters and symbols (including the additional vowels, typical conjunct consonants, Vedic characters and accent marking etc.) being available on the Computer. This would ensure that any Sanskrit matter can be keyed in.

While data entry for word-processing applications like letter correspondences. Publishing and printing etc. is relatively easily accomplished, it requiries additional prerequisites when a programming need arises. We are precisely concerned with this aspect in this paper. Here, we list basic 'processes' that we might need like exploding and imploding Sanskrit characters for any sort of string manipuation (like sandhi, sorting, augmenting, substituting or eliding characters) etc. for which we have to choose a Computer programming language. Now, we should ensure that our platform for Sanskrit processing (hardware or software) would allow the usage of Sanskrit characters in data or program files etc. GIST however, is compatible with many standard softwares like Lotus, Dbase, Turbo C, Turbo Prolog etc.

PROGRAMMING FOR SANSKRIT

BASIC FUNCTIONS

The next step after data entry is to choose a programming language. Here, the objective is to write a program to carryout

some of the functions listed above. Let's take *sandhi* for example. Here, your program should take two Sanskrit words as input, apply some rules on them and produce a combined word as output. But, to apply sandhi rules, we need to know the ending character of the first word and the beginning character of the second word. This will not be automatically known from the word if you consider what you actually key in while typing.

For example, if the word is *sudhī* and the second word *upāsyaḥ* is then the keys actually depressed for typing these words are respectively for the two words. Thus, to convert the *mātrās* into their respective vowels and combining the *halants* to the consonants (in some softwares to delete the *halant* in some cases) etc. we need to write general-purpose functions. The result of such a function would be to produce *s,u,dh,ī* and *u,p,ā.s,y,a,ḥ* for the two input words. Such general purpose functions are to be written into a library so that the same can be included in all further programs. These can also be used others without they having to rewrite them.

We can improve on these functions to include Vedic characters and accent character splitting etc. so that these functions will be quite powerful and have enormous applications. We, at C-DAC, are engaged in developing such function libraries, rulebases and databases for Sanskrit under DESIKA package, which will also be made available to interested users commercially. Further software development also is possible using DESIKA platform.

Now is the issue of applying some rule, say in this case, to begin with How are we going to 'code' this rule ? Here comes the issue of scheme to convert the *sūtras* into computaitonal form. This involves a proper analysis of the *sūtra* style, contents, number, arrangement, features, types, processes, metarules, conventions etc. We only enumerate here the salient points required to follow the current example. We first decide what rule to apply. (We don't need the *sūtra* in verbatim form at any stage at all. Except probably if we want to point the rule by which we arrived at the answer). The content on the *sūtra* is coded as follows:

Check if any additonal words are to be supplemented to the words of the *sūtra* (by adhikāra or anuvṛtti). If yes, supply those words. Then, by using interpretative rules, figure out

the operations enjoined by the *sūtra(s)* - here we also include the effect of the *sūtras* that lend words to this rule, by inheritance. This is typically in the form of specifying the LHS and RHS operand(s), the condition or environment in which they undergo one of the three processes of augmentation, substitution or replacement and elision. The particular process also is noted. As said earlier, general purpose functions for these processes with the number and type of operands as parameters are written and these are called with the relevant values for the parameters.

Thus, a rulebase can be built by using basic functions. We also need certain lists and procedures as databases like the case of *pratyāhāras*. The *Maheśwara sūtras* together with the procedure for forming *pratyāhāras* can help in listing all useful *pratyāhāras* and their expansion as a ready list (or database). Dhātupāṭha, Gaṇapāṭha, words of Liṅgānuśāsana are other typical databases that may be required. These are now created using the above m ntioned procedure.

Continuous application of further applicable rules at any stage can also be governed by the program, by calling different functions or routines written to execute individual rule information. Thus, a *subanta, tiṅanta, kṛdanta* or *taddhita* generation program would involve basic function libraries, relevant databases and rulebases, proper user interfaces for input/output etc. DESIKA has achieved this stage and is available for users.

Thus far, we concentrated on rule-based operations (mainly based on *Aṣṭādhyāyī*), but when we are to handle word analysis (or recognition), how do we go about ? This obviously requires applying the Pāṇinian rules 'in the reverse'. so to say. This is not straightforward as many different forward application of generative rules can cause the same end-result and backtracing must lead to multiple results. Here also, we need to prepare various judiciously devised databases (by close observation and analysis of word-types or paradigms) for bases and suffixes etc. One such attempt is available for study and scrutiny in DESIKA software. For convenience in machine handling, we may need coding (alphanumerically) of these details.

While we would very much wish to present illustraitons for all of these cases, we skip over for fear of space limitations in this article. However, a few references cited in the end throw more light on these factors.

Next in sequence are the *kāraka-vibhakti* mappings which act as the bridge between syntax and semantics as these specify the possible functional roles of words in a sentence. Beyond the stage of identifying the possible functional roles of words, we require the logical compatibilities and context matching. These necessitate an ontological approach to represent word-object meanings and here, we get sound theoretical basis in *Nyāya shāstra*.

Coming to these extra-grammatical factors, we need to treat primary and extended senses of words, the large class of qualifiers or modifiers, account for indeclinables used as fillers or for asthetics, style etc. and cover incomplete or wrong inputs in simple or complex constructions. The factors like mode, voice, tense, case, person, number etc. their agreement checks with the issue of multiple possibilities at each stage are also to be dealt with. Here, we look towards Nyāya and Mīmāṃsā concepts to understand the problems and subsequently devise suitable solutions.

Various computational techniques employed in AI are then evaluated to compare the efficacy of an ancient Indian method of viewing these situations and the western viewpoint. The resultant scheme would be applicable to most India (Sanskrit-based) languages and to that extent, remain language-independent. Of course, a major exercise of suiting the grammars of these languages into the common format and preparing of the databases etc. has to be carried out. However, the algorithm would be universal. This way, Sanskrit principles and langauge can contribute to Computer based studies. *Teaching and research* can then be the application areas where these methods can be tested for validaiton.

SANSKRIT FOR COMPUTERS

There is often a feeling or expectation (or is it intuitive pride ?) at lay person's level that Sanskrit is most suited as the 'Computer language', meaning, to operate the Computer better or efficiently, probably. This is the perception one gets from the media reports. However, if one considers what this means, if taken literally, we find that a Complier may have

to be written for use of Sanskrit instructions etc. Here again it may appear that a restricted or 'technical' version of Sanskrit (like *'sarala saṁskṛtam'* as termed by Drs. Vineet Chaitanya and KVR Krishnamacharyulu) can only be the candidate and not the commonly understood 'natural language' Sanskrit. The desirability or advantages of such a proposition also is open to more questions. Our opinion on this issue is brought out further.

While all the currently used Personal Computers are based on processors optimised for numerical operations (and are also being extended to character and string manipulation type of operations), it may be desirable to design a processor basically optimised for character string operations only. In order to draw up the specifications and requirements of such a processor, we got excellent clues from the meta-rule, meta-language and linguistic marker system of *Pāṇini's Aṣhṭādhyāyī* rules. Also one has to cater to the various paradigm types and processes involved in Sanskrit language word-formation (derivation), so that we will have in-built facilities for dealing with any string manipulation requirement of any Sanskrit-based Indian language in an elegant manner.

At the level of data types also, a holistic approach based on the Vedic system of study (with 14 Vidyāsthānas - 4 Vedas, 6 Aṅgas and 4 Upāṅgas) covering lexical, phonetic, syntactic, semantic, prosodic and accents, etymological and exegetic, socio-contextual parameters of word-attributes is to be taken as the foundation, so that meaning representation can be more successful and natural. Here, the Nyāya concept categories (covering both *pramāṇas* and *prameyas*) and Mīmāṁsā principles for sentencehood, discourse coherence, theory of knowledge, error etc. should form part of the semantic attributes of words (character string type). The processes of augmenting, replacing and eliding with the LHS, RHS, both or none as positions of operand(s) etc. and their denotation by rules and meta-rules can be also provided for.

The accent attribute also must be included to truly implement Pāṇini's system and this will benefit speech processing also immensely on a common pedestal. Here, we just fancied one scheme where the character set of Sanskrit being limited to 64, can be represented by 6 bits of each character (byte) and the two leading bits be used to denote

the four commonly required accent variations. This of course, would necessitate bit-level manipulation, but this can be handled by a simple encoder/decoder arrangement. This is something I loosely refer to as 'colouring' of data so that the Pāṇinian habit of achieving brevity and clarity can be inculcated into the system.

The concept of servile or mute consonants (somewhat similar to control characters) which provide additional information, but do not add to syllable count (and are subsequently dropped after their effect is taken), their position-related operations etc. employed like a magic wand by Pāṇini also need incorporation as in-built feature. The various types of chaining hierarchy, priority, inheritance etc. are to be available as basic operations, as also many procedures for representing the *Paribhāṣās* and some *saṁjñās*. The aspect of semantic criteria, including intention-related factors described by Pāṇini, also needs to be provided for. In short, instead of writing functions for each rule or type of rule in Aṣṭādhyāyī, we may think in terms of devising a special-purpose compiler dedicated for such functions meant for use in linguistic operations and come up with a 'linguistic workstation', catering to a variety of research issues in Computational Linguistics.

Regarding *pramāṇas*, some scheme should be devised for accounting for the source of knowledge like direct sensory perception (which faculty/sense organ), inference, verbal testimony (presumption, analogy, legends, non-comprehension, heresy) etc. These are vital in many cases and there must be a holistic picture (as close to a human system as possible) available for processing dynamically, during analysis. While frames, scripts etc. and many other conceptual structures can be employed for these, we want to look at the situation if one were to get a system 'made to order'. Also, the terminology is unambiguous in technical Sanskrit, which could be used at as deep a level as possible.

On the issue of Ontological categories (*prameya*), certain functions to associate the word to its meaning class in terms of substance and non-substance (attribute, activity and relation) and the multiple-class or level cases when viewed from different angles etc. are to be devised to dynamically be

able to create, view or modify associations etc. Here also, the prescriptions for essential and additional characteristics are to be dealt with integrally. Regarding activities, the associated functors alongwith their conceptual types may have a provision to be represented as attributes of the activity besides transitivity, mode, voice etc.

Provision for modifiers as basic functions of the Computer language may also be needed to take the effects of *upasarga* (preverb)s, accents, incomplete, multiple (sequential or simultaneous) and expected acts etc. Adverbs, adjectives of functors and their compatibility etc. are also to be taken care of. Regarding relations also, a multilayer representation should be possible dynamically. This should be integrated with lexicon, databases, rulebases etc. Type of relation, relata, parameter governing the relation etc. need to be handled. Types of input like plain, accented, prose, poetry, isolated sentence, discourse, *sūtras*, technical or classical literature of descriptive, prescriptive, narrative types etc. through written or spoken modes may also figure in the list of possible variations.

We may also study lexicographical and related issues to incorporate lexical building tools, databases interfacing, rulebase formulation and accessories etc. integrally. The linguists' and Computer Scientists' views may be elicited on these matters and if a core (broadly) team of a Sanskritist, a computer scientist and a Vedic scholar can go into these at length and in depth, much more may be understood on this topic.

This is essentially a wild thinking on a serious issue and I beg to be pardoned if silly thoughts are presented, but if any point seems worthwhile, it could be taken up for further study/scrutiny by experts. I invite comments from readers on all the issues raised so as to update myself on the many important aspects concerning a meaningful synergy between ancient Indian shastras and modern Computer technology. I do have faith in the synthesis materialising soon and contributing to better language processing in future.

References

1. "NLU System for Sanskrit" by B Sabarinath and Saji K David, M.SC (Comp. Sc.) thesis, Univ. of Kerala, Dec. 1991.

2. "Computer Processing of Sanskrit" by P. Ramanujan, in Proc. of Int'l Conf. on Computer Processing of Asian languages (CPAL-2), IIT, Kanpur, March '92.
3. "KR & NLP : An integrated approach based on ancient Indian sciences," by Vineet Jain, M.Tech. Thesis, IIT, Bombay, Jan. 1993.
4. Product Informaition (Brochure and Manuals) of DESIKA, C-DAC, Pune, June '93.

SANSKRIT AND COMPUTER

Vrashabh P. Jain

There could be three-point discussion on the topic "Sanskrit & Computer" viz. the first : how Sanskrit could be used for developing computational system, the second is to discuss the methodology of using Computer as a tool for the purposes fo the analysis, teaching, learning and researches of Sanskrit and the third is to focus on creating an inter-discriplinary area of research based on Sanskrit and Computer both. The present paper will give an introduction of above mentioned all the three points in short.

Computer was developed with a view that the information of the whole universe will be put in the machine and whenever anybody requires any piece of information already feed in, within no time it will make available that required information after processing. Western scientists are continuely working in this area of Knowledge Representation of Al since the infancy of Computer, they were using western logic, western mathematics, western philosophical systems, for developing their schemata, but till today they could not bring out any final methodology. As they go deeper, they find themselves more caught in, the problem becomes more complicated and they find no way to come out with solution. When we seriously look into all the western systems of understanding, we find that there is no inter-locking or inter-linkage among their western systems, every system claim itself as a complete system while it focusses only one aspect of the problem. Every objects of universe is multi dimensional, to view anything in one aspect is to see that in part and not in totality. Our Ancient Indian Systems of the analysis of universe and understanding are such systems which are interlinked with each-other, which are complementary to each other, which need each other to explain each other. Hence every system of Indian Thought is a co-system of the other system, and they are the sub-systems of the whole system, that is why they form an Unilateral Model. If anybody wants