Of course, all of this is on the assumption that you can recognise Kṛttikā in the sky! Kṛttikā is a cluster of six stars well known as Pleides. There is an interesting Hindu mythological story about this. It appears that the six wives of six of the seven universal Ṛṣis of this age received the Energy of Lord Śiva, through a miraculous happening involving the involuntary intervention of the Lord of Fire. They conceived and brought forth six children all of whom combined into one child with six faces. This is the deity Ṣaṇmukha. The six husbands cursed their wives to become a cluster of stars and that is the Pleides cluster of six Kṛttikā sisters. Because they nursed Ṣaṇmukha as a child, the latter is also known as Kārtikeya. Western mythology talks of this cluster as the seven sisters of Greek mythology. The cluster is actually made up of a large number of stars, the number going up to hundred with more and more powerful telescopes.

There is another subtle correction traditionally applied to the 27 formulae which goes as follows:

jyeṣṭhādhīnām atikramya revatyādīn anāgame।
ārdrādīmśca samam dṛṣṭvā nava-ṣaṭ-dvādaśa-krama(śa)ḥ॥

This means: The nine asterisms starting from Jyeṣṭhā must be seen *past* the meridian; the six asterisms starting from Revatī must be seen *before* they cross the meridian; and the twelve stars starting from Ārdrā must be seen exactly when they are *on* the meridian — for the purpose of the above formulae!

Such are the details of the interesting formulae and the patterns by which we may recognise the 27 asterisms which have been *traditionally current.*

RELEVANCE OF ŚĀSTRAS FOR NATURAL LANGUAGE PROCESSING

P. RAMANUJAN

0. ABSTRACT

In this paper, the issues in Natural Language Processing (NLP) which could possibly benefit from Sastraic studies are discussed. Only the basics are touched upon and a few practical implementation issues are detailed. The DEŚIKA package developed at Centre for Development of Advanced Computing (C-DAC), Poona is referred to throughout to clarify/support discussions.

Word, sentence and discourse (Syntactic, Semantic and Pragmatic) levels are considered. Knowledge base consisting of data and rules are covered. Issues regarding Generation and Analysis modes of plain and accented Sanskrit input processing are briefly explained. *Subanta, tiṅanta, kṛdanta* and *taddhita* forms etc. are discussed. Implementation of simple sentence analysis is described.

Semantic analysis includes generic and specific syntactico-semantic mappings, activity and conceptual classification, ontological characterisations, sentence coherence factors etc. drawing upon from the Śāstras. Computer outputs from DESIKA package are available.

1.0 INTRODUCTION TO SANSKRIT AND NLP

Ever since Comparative Philology and Indology became subjects of serious and significant studies (for over a century now), Linguists have paid attention to Pāṇinian

grammar in ample measure and Navya Nyāya theories of meaning, sentential import etc. to a good extent. With the advent of Computational methods of analysis, Computational Linguistics has become an area of advanced research.

In Artificial Intelligence (AI), issues like Processing of Natural (Conversational) Languages, Representing Knowledge in computers (particularly Inference) and modelling common sense/contextual knowledge are the frontier areas of research. Here, ambiguity is 'the' issue while interpreting (and, of course, during generation also). Added to this is the factor *vivakṣā* ('speaker's intention') for the determination of which hardly any help is available. Thus, this is essentially multi-disciplinary involving Cognitive Sciences as well, since language comprehension requires human competence at lexical, syntactic, semantic, phonetic, prosodic, cognitive and socio-contextual levels. As an example for a computer to 'understand' a simple sentence like "the master teaches scriptures to ascetic disciples" all the factors aforesaid are to be available in the machine.

In Indian Śāstras (scientific treatises), these factors are dealt with integrally and the result is the formulation of a comprehensive system of language description for correct usage. This system has Vedas at the nucleus and a host of auxillary sciences to assist their understanding and preservation. These are really devices for ensuring distortion-free transmission over generations, through oral tradition. This system has remarkably served the cause for millennia, and that is precisely why there is interest in modern scientists working in AI about these literary traditions (details later).

Currently, the three major Śāstras, viz. Nyāya, Vyākaraṇa and Mīmāṃsā are studied with great interest in search of clues regarding the above mentioned issues. Some of the recent efforts in this direction are: Pāṇini's grammar (auto-semantics) as cognitive knowledge structure in 'Simurg' project in Germany, two-level morpho-phonology of Sanskrit in Finland, Morphological analysis of Sanskrit by computer in Netherlands, Paninian database project in U.S. and National Language Understanding (NLU) systems based on *śābdabodha* theories described in the Śāstras, in India. The C-DAC, Pune, has developed GIST (Graphics and Intelligence based Script Technology) card which provides facility for processing linguistic data in any of the Indian scripts (and also many foreign scripts). DESIKA, a package developed for study of Sanskrit, works in GIST environment (details later).

2.0 ISSUES IN NLP

Many of the problems confronting NLP boil down to three of the central issues in AI, i.e., *representation, reasoning* and *recognition.* It is essential to devise means to tackle these issues as AI seeks to give a computational account of intelligent behaviour, which should include understanding communication. Also, much of world's knowledge is set down in natural language and if it is to be used by artificial entities, it should be converted to forms amenable for artificial manipulation.

3.0 CONTENTS AND RELEVANCE OF ŚĀSTRAS

Traditionally knowledge is classified topically into subjects called Vidyāsthānas. These include four Vedas (scriptures),

their six limbs (Vedāṅgas) and four supplementaries (Upāṅgas). While Vyākaraṇa is a limb, Nyāya and Mīmāṃsā are supplementary to Vedas. Vedas are the treasure-house of all knowlege and are proverbially infinite and eternal. They are chiefly classified into *Ṛk, Yajus, Sāma* and *Atharva* Vedas. Understanding them thoroughly is not possible without the six limbs which deal with the factors as shown below in the order (name — meaning — linguistic factor dealt — limb type, explanation).

Nirukta, etymology and exegesis (lexical) — Ears, provides a repertoire of words (accented) or philological explanation of difficult Vedic words; Vyākaraṇa, grammar (syntactic) — mouth, helps generate innumerable grammatically valid word-forms from a finite set of roots through rules. Śikṣā, science of pronunciation (phonetic) — nose, classifies sound phonetically and defines pronunciations and euphonic combination including accents for the character set of the language; Chandas, metrics/prosody (prosodic) — feet, defines intonation structure for speech to communicate emotions appropriately; Jyautiṣa, astronomy (temporal and spatial) — eyes, define the suitable occasions for proper results; Kalpa, ceremonial (practical) — hands, prescribes rituals and rules for ceremonial and sacrificial acts.

Among the four supplementary sciences, Nyāya deals with semantics by a system of logical compatibility and validity rules; Mīmāṃsā gives guidelines for interpreting Vedic texts and deals with discourses and philosophy; Purāṇas detail the mythological and spiritual aspects of life while Dharmaśāstras lay down moral codes of rectitude.

Of these, the themes which are found to be directly relevant and readily usable as yet, are the structure of

grammar of Pāṇini as a rulebase for natural language generation (and syntactic analysis) and the method of *śābdabodha* (semantic extraction) described in Vyākaraṇa (at word level), Nyāya (at sentence level) and Mīmāṃsā (at discourse level). However, when Vedic analysis is to be done, which is of paramount importance to clearly understand the Sastraic system, the others are also inevitable.

3.1 The Vyākaraṇa Śāstra

This science with rules by Pāṇini contains well-structured description of Sanskrit grammar and deals with word-level and sentence-level syntactic aspects. Both plain and accented forms are exhaustively treated. Besides the rules called *Aṣṭādhyāyī,* there are databases for *gaṇapāṭha* (of nominal stems), *dhātupāṭha* (of verbal roots), *liṅgānuśāsana* (gender determination) and *śikṣā* (phonetics). The theory of *kāraka* (functional relationship) relating the case structure of syntax and the denoted objects is a major outcome of universal application to natural languages. The significant aspect of the import of a sentence according to scholars is normally the *activity* denoted by the verbal root: *dhātvartha-mukhya-viśeṣyaka-śābdabodhaḥ.*

3.2 The Nyāya Śāstra

This science with rules by sage Gautama deals with *padārtha vibhāga* (conceptual classification of things) as *pramāṇa* and *prameya* (means and objects of knowledge). Theories of validity and error are also covered. A model structure of technical language for unambiguous description (Navya-Nyāya) is a by-product. The significant aspect of the import of a sentence according to scholars of this

Śāstra is normally the *word in nominative case* (agent) of the action denoted by the sentence: *prathamāntārtha-mukhya-viśeṣyaka-śābdabodhaḥ.*

3.3 The Mīmāṃsā Śāstra

This science with rules by sage Jaimini establishes principles for understanding the contents of Vedic passages by maxims and pragmatic reasoning. Contextual factors and common sense aspects are formulated with hierarchy of knowledge-sources for conflict resolution and disambiguation. Various means of practical wisdom are a result of these formulations. Moral codes and jurisprudence are derived from some of these in Indian culture. The significant aspect of the import of a sentence according to scholars of this Śāstra is normally the *bhāvanā* (*intention* or force - typically in injunctions or commands): *bhāvanā-mukhya-viśeṣyaka-śābdabodhaḥ'.*

4.0 COMPUTATIONAL KNOWLEDGE BASE

To be able to utilise knowledge contained in natural language in free form within a machine, certain formats are to be devised and the knowledge is to be represented suitably in a machine-understandable form. *It is to be noted that the machine can only process 0's and 1's and any information is to be made available in this form ultimately.* To achieve this conversion of free form in natural language to formal machine form, many techniques are available. The Śāstras can also assist in devising novel means for the purpose. Thus, any piece of information pertaining to knowledge of any type, has to be stored in proper form and this stored and formated information

is termed 'knowledge base'. This is usually divided into two parts;' i.e., the database (information processed dynamically or variable) and the rulebase (static information which is applied on the database). So, Kośa (lexicon) and Sūtra (rules) traditionally (*aṣṭādhyāyī jaganmātā amarakośo jagatpitā*) correspond to the two parts of knowledge base. (A knowledge base usually exists in alphanumeric form. of character strings and numeric codes).

4.1 SYNTACTIC (GRAMMATICAL) DATABASE

As said earlier, the dynamically processed (or variable) information could consist of the fundamental lexical units of language which are processed for generation or analysis. Thus, *prakṛti* (stem or root) and *pratyaya* (affix = prefix, in-fix or suffix) can be the basic data units. Lexicon like *Amarakośa* can be thought of as the database for nominal stems and indeclinables and *dhātupāṭha* for verbal roots. We also treat the various affixes as data and store them accordingly (*Pratyaya Kośa*).

4.11 Prakrti Kośa (BASE LEXICON)

This lexicon would have information fields of *prātipadika* (nominal stem), ending, gender(s), declension type (explained later), derivation type, meaning, concept type etc. for all the words of say, *Amarakośa,* typically. Of these fields of information, the word alone need be in character-form while the rest of the details could be in a suitably 'coded' form. Similarly, for verbal roots, the *dhātupāṭha* could be had in fields of roots with '*It* (s)' as a character field and information regarding the *gaṇa, dhātu* and *it*

svara, padī etc. in a 'coded' form. *Avyayas* (indeclinables) are also 'coded' about their type (like *upasarga, gati, nipāta, karmapravacanīya, kṛdanta, taddhitānta* etc.), formation, base and suffix if any, meaning category etc. One way of 'coding' attempted in DESIKA would be discussed in implementation.

4.111 Amarakośa (VOCABULARY)

As an example of base lexicon, we could have a database of the *Amarakośa* in its original form (i.e., verses) and have programs to interpret and process the data or we could also have a hybrid scheme with both coded and original forms and provide quotation from source with reference, as needed. Here, the technical definitions (*saṃjñā*s) of the author can be interpreted by program functions, e.g., *rūpabheda, sāhacarya, viśeṣa vidhi* etc. by condition checks, *triṣu, dvayoḥ, seṣārtha, tvanta, athādi* etc. by interpreting functions. We can also analyse the text using these meta-rules or definitions. Other Kośas (lexicons) can also be similarly handled.

4.112 Gaṇapāṭha (LIST OF NOMINAL STEM CLASSES)

Among the nominal base lexicons, Pāṇini's *gaṇapāṭha* lists various stems with a certain syntactic property, particularly, regarding *taddhitas* (secondary derivatives) etc. These happen to be the data when we program *taddhita* generation or analysis. Here, membership of a particular list and its characteristics are of interest and this information can be easily coded.

4.113 Dhātupāṭha (LIST OF VERBAL ROOTS)

Pāṇini's *dhātupāṭha* is the source of verbal root data. We can code information regarding *gaṇa, padi, it karmitva, iḍāgama, dhātu svara* and *it svara* etc. explicitly or derive them using *Aṣṭādhyāyī* Sūtras by program functions. Other *dhātupāṭhas* can also be used similarly. Here, roots and their meanings could be in character form while the rest of the details can be in numeric code form. The effect of *upasarga*s (preverbs) on root meanings would also have to be stored as data.

4.12 Pratyaya Kośa (AFFIX LEXICON)

This includes nominal declensional, verbal conjugational and modal, primary and secondary derivational affixes, feminine suffixes etc. in a categorised manner. Here also, the affixes are character fields while their characteristic parameters could be coded. Semantics are also included in terms of relationships, properties etc. For example, '*sup*' (nominal declensional) suffixes denote case relationships and number, '*tiṅ*' (verbal conjugational and modal) affixes denote temporal, modal, active, personal and numeric relationship of relata with the verbal root meanings. *Kṛt* and *taddhita* suffixes (primary derivatives) denote *kāraka* (functor), *śeṣa* (residual or generic) and patronymic relationships etc. as specified in *kāraka, vibhakti* and *taddhita prakaraṇa*s (sections) in *Aṣṭādhyāyī*.

4.13 OTHER (SEMANTIC) DATABASES

Besides grammatical data, other aspects of linguistic data mentioned earlier are also to be suitably available

in the machine. Some of these are detailed in the following sections. *The main point here is that the Sastraic system is exhaustive and comprehensive in nature such that by appropriately representing its multi-dimensional character, we can build a good aid to its thorough study through the advanced computational processes. It will also help define a knowledge representation system based on ancient Indian tradition and comparatively study the modern developments in its light.*

4.131 Padārtha Vibhāga (CONCEPTUAL DATA)

The conceptual categories and ontology established in Nyāya Śāstra could be stored as semantic or logical database, in the form of frames etc. The base lexicon can be integral with the semantic features as well (as the manual dictionaries contain) or there can be a separate semantic lexicon, in which case, the linking of the syntactic and semantic lexicons has to be provided.

4.132 Śāstrāntara Kośa (APPLICATION DOMAIN DATA)

At discourse level, for different applications, contextual and domain-specific information about the concerned field would be needed. Thus, a database of the particular branch of study is to be established with links to syntactic and semantic lexicons. Mīmāṃsā Śāstra happens to be very important from the point of view of explaining the concepts in contextual and pragmatic reasoning. The formulation of its database is thus essential. Similarly, preparation of databases for other Vidyāsthānas also could be undertaken.

4.133 Vaidika Kośa (VEDIC DATABASE)

For the purposes of Vedic processing, certain additional databases pertaining to accented words and text would be needed. Thus, the Nighaṇṭu of Yāska (which is a database of assorted Vedic words), the Nirukta (which explains the Nighaṇṭu etymologically) in terms of derivative processes, the parameters of pronunciation and phonetics given in Śikṣā works for the character set of the language, the Śākhā (Vedic branch) specific special characters etc. from Prātiśākhya works and the entire Vedic text in *padapāṭha* form as the 'fundamental corpus' are required to be established.

4.134 Sāhitya Kośa (LITERARY DATABASE)

For the study of classical Sanskrit treatises like *Rasagaṅgādhara, Kāvyaprakāśa* or works like *Śiśupālavadha, Anargharāghava, Raghuvaṃśa* the respective texts will have to be 'keyed-in' and a database created. Here, through a lexical update program one can only update those entries that are not already part of the lexicon, dynamically (even while analysing), i.e. on-line.

4.135 SECONDARY DATABASES

Certain additional databases in the form of technical terms, definitions, tables of correspondence and mappings, query words and their expected answer types, statistics, links between various databases, schemes of alphanumeric coding for the different parameters of the database etc. e.g., all *śāstrīya saṃjñās* (*pāribhāṣika śabdas*), *subanta*

paradigm types and so on are also to be prepared. These are mainly guided by programming considerations.

4.2 RULEBASE

The rulebase is a set of linguistic instructions for processing the input data. This could be in the form of 'if (*condition*) then (*action*)' rules, or a simple or complex network linking input and output states or certain concepts or descriptions. Most of syntactic processes may be of rules form, the semantic relationships being graphically or conceptually defined. .Only empirical principles may be all that is possible in pragmatic levels of processing.

4.21 GRAMMATICAL RULEBASE

Grammar of Sanskrit deals with syntactic details at word and sentence levels, mainly for the spoken form. There is one-to-one correspondence between spoken and written forms. *Svara*s (accents - different emphases) of sentence and euphonic combination of words are also covered. These are available from *Aṣṭādhyāyī* arranged topically and process-wise. The processes are basically *ādeśa* (substitution), *āgama* (augmentation) or *lopa* (elision) at lexeme, phoneme or morpheme levels.

4.211 Aṣṭādhyāyī

This is the well-known descriptive rulebase of Sanskrit grammar by sage Pāṇini. There are eight *adhyāya*s (chapters) with four *pāda*s (quarters) each and having a total of about 4000 *sūtra*s (aphorisms). The rules are extremely brief, require meta-rules to interpret and fill ellipses and

are very closely-knit (in nested chaining-form). The rules are of *saṃjñā* (definitive), *paribhāṣā* (interpretative), *adhikāra* and *anuvṛtti* (jurisdictional), *vidhi* (operative), *apavāda* (exceptional), *niyama* (restrictive) or *atideśa* (extensive) types. These rules can be effected by suitable functions in program and the process carried out. The trace of the process of generation of recognition is also possible to be given by the computation process.

4.212 Liṅgānuśāsana

The rules pertaining to determination of gender of words are listed in Pāṇini's *Liṅgānuśāsana* which has a simple rule — exception - extension sequence and contains 191 rules in all and lists about 1200 nominal stems. *Amarakośa* also has these factors covered in *liṅgādi-saṅgraha-varga*. With these rules programmed, the user can just specify a *prātipadika* (nominal stem) and get the valid declensions through the system itself, including multiple genders.

4.22 OTHER RULEBASES

The semantic, cognitive and socio-contextual aspects are not easily amenable for simple formulation in the machine. Here, the concerned Śāstras also have evolved and adopted appropriate techniques which are very relevant to AI and NLP. These include classification of things by characteristic parameters, logical compatibility criteria, truth and error value definition and philosophy, sentential coherence issues, propriety conditions etc. detailed thoroughly in Nyāya

and Mīmāṃsā Śāstras. These are also to be stored suitably in the machine for semantic and contextual analysis, conflict resolution and disambiguation. Common sense formulation, world knowledge representation etc. need these details badly.

5.0 PROCESSING STAGES/LEVELS

We now detail certain stages of computational processing of linguistic data pertaining to case-based languages like Sanskrit. We consider both generation and analysis (recognition) at word, sentence and discourse levels. Parsing a given sentence or creating a sentence to denote indicated information is the test of machine understanding of natural language. *This understanding may be demonstrated by query processing, voice change or paraphrasing the input sentence* for a parser. Conversely, parser output should be input to the generator to obtain the input sentence.

5.1 WORD LEVEL

Generating inflectional natural language words and recognising finished words in terms of base and affix (including multiple possibilities) require computational knowledgebase aforesaid and a proper algorithm (step-by-step method) for arriving at the output from the input, systematically. This procedure would then be capable of capturing the traditional expertise in its exhaustive form for all future uses and refinements. Thus, 'knowledge' can be preserved *in contra-distinction to 'data' or 'information' storage and retrieval.*

5.11 Niṣpattiḥ (GENERATION)

For generating natural language words, we need as the input for base, a nominal stem or verbal root and for affix, a specified semantic feature like functional (case) or personal relationship etc. For derivative forms, the significant criteria are to be specified. Here, the machine offers invaluble help in listing all possible choices that can be indicated by using suitable databases and structures or formats. All the rules involved in the process of generation together with a 'trace' of all intermediate results on application of each rule or condition is obtainable. This feature would be invaluble in academic efforts like study, research etc. We now consider various word types in Sanskrit.

5.111 Subanta (NOMINAL)

Nominal declensions take archetypal terminating suffixes denoted by the siglum '*sup*' and are inflected into eight possible cases and three numbers. Thus, the nominal stem and the desired case and number are to be input for generation. Here, ending character (vowel or consonant) of the stem and its gender are the two factors that influence the way the archetypal '*sup*' suffixes get modified. Other parameters like pronouns, certain derivative types and specific syntactic criteria also cause special modifications to the suffixes. Thus, the declension is according to the applicable paradigm type. The process of internal combination of stem and suffix also may entail certain modification (e.g., cerebralisation) in inflected form. It is thus possible to establish all distinct declensional paradigm types beforehand by consulting the rulebase and use them in generation.

We have identified 327 distinct paradigm types. Works like *Śabda Ratnāvali* could be generated by the machine automatically, at will, with explanations.

5.112 Tiṅanta (VERBAL)

Verbal declensions depend on root parameters like the conjugation (or group), indicatory characters, accents, mode, voice, tense or mood, person and number. They may also be affected by preverbs, if any. Here, the archetypal affixes are denoted by the siglum '*tiṅ*'. The input is a root with indicatory characters, alongwith the desired choice of preverb(s), mode, voice, tense/mood, person and number. Here also, generalisation of the modification of the archetypal affixes into paradigm types may be attempted theoretically. However, our studies have shown not much savings in efforts on this count as the diversity is high (1213 paradigm types for around 2000 roots). It is thus best generated, *ab-initio,* using rules. Works like *Dhāturūpakośa* could be generated by the machine automatically, at will, with explanations.

5.113 Kṛdantas (PRIMARY DERIVATIVES)

For *kṛdanta*s, the choice of verbal root, preverbs, if any and the desired *kṛt* suffixes (with *anubandha*s) are input. The effects of these are applied through rulebase and the necessary processing is carried out (including *sandhi*s). A paradigm type analysis of *kṛt* suffixes on semantic basis would help represent meanings for further stages. Thus, the meaning could also be indicated instead of the desired suffix (action sequence, certain functor,

relationships or material properties etc.). These are highly expressive word forms covering participial, infinitive, substantive, adjective, gerundial types etc. Even works like *Kṛdantarūpamālā* can be generated automatically when the system is completely developed, with the added advantage of providing 'trace' of the rules involved and intermediate forms.

5.114 Taddhitas (SECONDARY DERIVATIVES)

For *taddhita*s, choice of nominal stem (from lexicon, *gaṇapāṭha* etc.), and desired *taddhita* suffix are input. Here also, grouping of *taddhita* suffixes on functional basis can be carried out and this information is used to select the desired form. List of *taddhita* forms are thus generated.

5.115 Samāsa (COMPOUNDING)

To generate compound word-forms, the constituent word-bases and the semantics (meanings) of compounding could be the input. The semantics can be mapped to specific relationships (functor or residual) of compounding. Thus, the object denoted by the former, latter, all or external word(s) may normally map to *avyayībhāva, tatpuruṣa, dvandva* and *bahuvrīhi* compounds. Thus, by applying the rulebase, compound words can be formed from components.

5.116 Vaidika (ACCENTED INPUT)

In Vedic processing, Saṃhitā (continuous text or prose), Aṣṭavikṛtis (*krama, jaṭā, ghana, ratha, dhvaja, daṇḍa, rekhā, mālā*—the eight fold combinatorial patterns) or Pañcasāra-

varṇa-krama (complete characterisation of each Vedic syllable with 26 attributes — 8 each for vowel and consonant parts and 10 for accent part) from *pada pāṭha* (single words) are used as input. These processes involve accented *sandhi*s using the concerned Prātiśākhya and Śiḱṣā, besides grammar rules. Thus, generation of Vedic forms does not mean taking a 'base' for declension in the conventional sense.

5.117 Sandhi (EUPHONIC COMBINATIONS)

This takes two or more words for input and combines two words at a time to produce the combined output form. Internal and external *sandhi*s for vowels and consonants (using one or more rules in stages) giving mandatory and optional forms are defined.

5.12 Vyutpattiḥ (ANALYSIS)

Word-level analysis refers to recognition of a given finished word in terms of all possible (grammatically valid) identifications of base and affix combinations. *This is the single most important feature of the parser for Sanskrit which would have immense utility.* This could also take accented input. Thus, syntactic analysis would indicate all distinct senses of a given word for further determination of 'intended' sense under given context by semantic and pragmatic analyses. Few examples are *rāmaḥ, aśvaḥ, aruṇaḥ, vṛkṣe, tasya, gacchati, bhavati, yācate, mātaḥ, yānti* (both noun and verb), *yajeta* (Parasmai and Ātmanepadi), *vakṣyati* (*vaha* and *vaca* as roots), *gate* (all numbers), *naraḥ* (*ṛkārānta* and *akārānta*), *rāmāḥ* (masculine and feminine), *te* (8

possibilities), *mā, gātum, viśvasya* (noun and indeclinable), *asmi, āha* (verb and indeclinable) etc.

The recognition of a word requires that the base and affix are available in the lexicon in the machine. Only the nominal bases are too numerous to include exhaustively and hence, an on-line lexical update facility for Prātipadikas can be provided. The program would prompt the user to exercise the option to update the lexicon with proper choice of stem, gender and paradigm type as required, when a particular word is not identified. Thus, any real text could be given to the program for analysis, without concern regarding the lexical content.

5.2 SENTENCE LEVEL

Generating sentences or analysing them syntactically and semantically is involved in this stage. Here, the individual words are to be processed in a sequence defined by the input. The sentential import may introduce modifications to the 'sigma' of individual (component) word level results and these factors are quite tricky to handle. This calls for careful choice and design of guiding principles.

5.21 GENERATION

Sentence generation requires detailed inputs. The user has to be given proper alternatives to choose from at various stages for different types of sentences. In our analysis, few hundred types with well over a thousand sample sentences collected from standard works like *Vaiyākaraṇa Siddānta Mañjūṣā, Vaiyākaraṇa Siddhānta*

Kaumudī, Śabda Śakti Prakāśikā, Bhāṭṭa Rahasyam and *Śābda Taraṅgiṇī* are used as the basis. The parameters for classification could typically be: type, accent, voice, mode, mood, tense, person, number, construction, functors, beneficiary of action, verbal root, meaning, transitivity etc. Besides these, aesthetics, ellipsis, stylistics etc. are also involved.

The algorithm involves selecting the activity to be denoted, to begin with (for active type of sentences, of counsel). Then to root, pre-verb (if required), mode, voice, tense, person number, beneficiary of action, functors, their base, gender, number etc. at word level are selected. Then, for sentence level, construction type, accent, ellipsis, aesthetics etc. are selected. The appropriate rules of generative grammar are then applied to choose the suffixes etc. for each input parameter and the individual word forms are generated. The sentence level requirements are then checked for compliance and the words are strung into a sentence by sentential processes (preferred order, *sandhi* etc).

5.22 ANALYSIS

Here, each word is first syntactically analysed and all possible recognitions for each word are collected. Then, with verb as the pivot, unique functional roles are assigned for each word using *ākāṅkṣā* and *yogyatā* (*kāraka* requirements and conceptual type compatibility checks). All verbs are mapped to their meaning types, *kāraka* specifications (mandatory, optional and inhibitory) and concept type compatibility criteria. Presence and capability of candidates for each necessary role of functors in the sentence guide

the process. This also helps in identifying incomplete or incompatible sentences. When all the words of the input sentence are uniquely assigned distinct functional roles, the sentential *import* is output by a suitable choice of *śābdabodha* (paraphrasing) form.

5.3 DISCOURSE LEVEL

Over and above the sentence level, pronominal and other forms of back references, anaphora, ellipsis etc. are to be particularly addressed at this stage.

5.31 GENERATION

Multiple sentences are to be generated here and an episodic structure has to be built up dynamically. The reference factors etc. are included suitably and contextual criteria enumerated in Mīmāṃsā are utilised as also common sense issues from *laukika nyāya*s. This is too advanced a stage to offer any concrete points as of now.

5.32 ANALYSIS

Knowledge base of local domain and formulation of disambiguation criteria, common sense etc. are required here. In our system, a typical *yajña* (sacrifice) has been described as word knowledge and basic Mīmāṃsā maxims are emulated to facilitate study of the issues involved, in depth. In both sentence and discourse levels, query processing is possible to enhance the quality of understanding the system processing. Voice change and precis output are other yardsticks to measure the extent of machine

understanding. Also, in analysis, the results can be input to generation module for verification and *vice-versa*.

6.0 IMPLEMENTATION SO FAR

The various databases and rulebases mentioned in the preceding sections have all been incorporated, some to a preliminary level and some to a fairly good extent. The main point is that many of the thoughts presented here are clearly realisable as some amount of practical solution to these are available. Due to space constraints, all the details of our system are not accommodated in this paper, but are available on request. A brief information brochure on DESIKA is appended with this paper. Kāraka prakaraṇam (Functor - Case Mapping), Vibhakti prakaraṇam (Sub - Functors Mapping), Dhātvartha viśleṣaṇam (Verbal activity analysis), Padārtha vibhāga (Conceptual Categories) and Vākyānvaya (Sentence & Discourse Coherence) are some important features that are incorporated.

7.0 CONCLUSION

We have made a beginning in seriously studying the utility of Śāstras for NLP in our system and the result is encouraging. We appeal to all Sanskritists, Linguists and Computer scientists to support and encourage us by trying our method (and system) to enrich and revive Sastraic studies from computational angle.

REFERENCES

"Desika - A NLU System for Sanskrit", Project Report, Centre for Development of Advanced Computing (C-DAC), 1992.

Devasthali, G.V. *Anubandhas of Pāṇini,* Poona, 1967.

Katre, S.M. *Aṣṭādhyāyī of Pāṇini,* Roman transliteration and Eng. tr., Delhi, 1989.

Palsule, G.B. *The Sanskrit Dhātupāṭhas: A critical study,* Poona, 1963.

Pandit, M.D. *A study in compound word-forms,* Poona 1961.

— *A study in non-compound word-forms,* Poona, 1963.

Ramanujan, P. "Computer Processing of Sanskrit", *Proceedings of Conference on Computer Processing of Asian Languages* (CPAL-2), I.I.T., Kanpur, 1982.

— "Computerisation of Vedic texts", *National Conference on Vedas and Shastras,* Tirupati, 1992.

— "Mastiṣka Yantre Śābdabodha Pratirūpaṇam", *National Conference on Vedas and Shastras,* Tirupati, 1992.

DESIKA-NLU SYSTEM FOR SANSKRIT

General Description

Sanskrit is an ancient Indian language renowned for its highly structured grammar described by Sage Pāṇini, millennia ago. Even in modern times, the language has evoked keen interest among Linguists and Computer Scientists for possible clues to Natural Language Understanding (NLU) issues. DESIKA serves the need for an authentic Computer-based package for its study.

DESIKA is a comprehensive package for generating and analysing Sanskrit words. It caters to different user communities like Academicians, Students, Researchers, Linguists, Computer Scientists etc.

DESIKA

सन्नन्तः कर्मणि

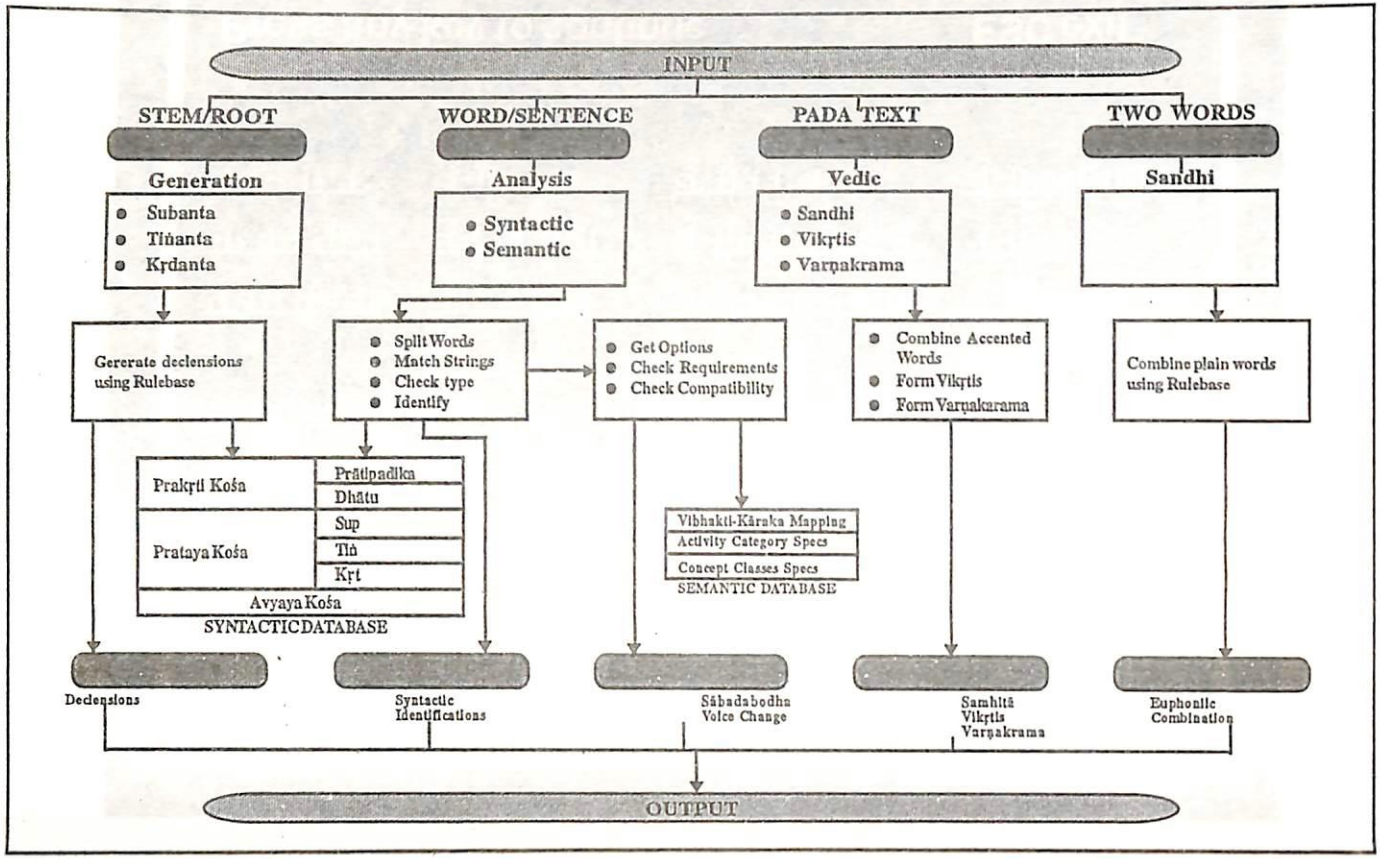
भू धातुः भ्वादि गणः परस्मैपदी अकर्मकः सेट्

आशीर्लिङ् लकार-

	एकवचनम्	द्विवचनम्	बहुवचनम्
प्रथम पुरुषः	बुभूषिषीष्ट	बुभूषिषीयास्ताम्	बुभूषियीरन्
मध्यम पुरुषः	बुभू षिषीष्ठाः	बुभूषिषीयास्थाम्	बुभूषिषीध्वम्
उत्तमपुरुषः	बुभूषिषीय	बुभूषिषीवहि	बुभूषिषीमहि

Press any key to continue ESC exit

Functional Block Diagram of DESIKA



MODULES

Generation

★ Has a nominal lexicon of 3500 words selected from *Amarakośa* and *Liṅgānuśāsana,* updatable on-line.
★ Covers ALL nominal declension paradigm types.
★ Covers ALL verbal roots of Pāṇini's *dhātupāṭha,* in all tenses/moods (including 'let'), verbal, causative, desiderative, intensive, frequentative and reflexive modes, and in active and passive/impersonal voices.
★ All *avyaya* types covered.
★ Important *kṛdantas* included.
★ Output in conventional format in Devanāgarī script (other Indian scripts selectable)

Analysis

★ ALL grammatically valid syntactic identifications of plain or accented, input word (non-compound, currently) or sentence, output in a suitable format.
★ *Kāraka* relationships are determined based on relevant mapping rules.
★ Sentence coherence based on *śābdabodha* theory and output according to Naiyāyika, Vaiyākaraṇa or Mīmāmsaka formats.
★ Output is **paraphrased** or **voice changed version** of input in Sanskrit.

Vedic

★ Covers *svara* (accent) *sandhis* (for Taittirīya Kṛṣṇa Yajur Veda currently using Prātiśākhya and Vyāsa Śikṣā)
★ Generation of Samhitā and Veda Vikṛtis (combinatorial patterns) like Krama, Jaṭā, Ghana etc. from Pada text as input.
★ Varṇakrama (exhaustive characterisation of each Vedic syllable) for Pada text.
★ Output in a suitable format in Devanāgarī script with accents' marking.

Sandhi

★ Combines plain or accented words and outputs result with the relevant rule/operation.

Features

- ★ Modules for Generation, Analysis, Vedic, and Sandhi.
- ★ Serves as a PANINIAN platform for computational linguistic studies/research.
- ★ Substrate for development of similar packages for Indian languages.
- ★ Useful in development of knowledge bases.
- ★ Based on *ab-ultimo* morphological analysis.
- ★ Indispensable for Vedic (accented sacred texts) analysis and preservation.
- ★ Suited for a tutorial on Pañcasandhi, Ṣaḍliṅgi, Kāraka and Tiṅanta sections (prakaraṇas) of *Vaiyakaraṇa Siddhānta Kaumudī*.

Knowledge Base

Database is 'coded' parametrically through an optimal alphanumeric code, for computational use.

Subantas	Ending, Gender, Paradigm Type
Avyayas	type, meaning
Tiṅantas	Roots, Conjugation, indicatory characters, usage, union-vowel, transitivity, accents, meaning
Kṛdantas	Root, Suffix, Semantic Type
Sandhis	Origin, Internal/External effort, etc.
Vedic accents	Udātta, Anudātta, Svarita, Pracaya
Vedic characters	Anusvāra, Svara-bhakti, Raṅgapluta

Rulebase consists of the relevant *Aṣṭādhyāyī* sūtras classified according to definition, operation (including exception and extension) and interpretation. In Vedic processing, Prātiśākhya and Śikṣā rules are also included. So far, over 500 rules of *Aṣṭādhyāyī* have been covered.

System Requirements

★ IBM compatible PCs
★ GIST-9000 add-on card
★ MS-DOS Ver 4.0 or above.

KNOWLEDGE REPRESENTATION THROUGH *ŚĀBDABODHA* AND SANSKRIT GRAMMAR

D. K. SUBRAMANIAN

Introduction:

There has been a lot of interest these days about the use of Sanskrit language in computers. Many people have claimed that Sanskrit is an ideal language for computers. This has led to a lot of controversies and confusions. Some people think that Sanskrit will replace Fortran or Cobol as a computer language. There was a comparison regarding this in a paper presented during the First National Seminar on "Knowledge Representation and Samskritam" held at Bangalore in 1986. It has been argued that many mathematical expressions cannot be expressed clearly in Sanskrit.

A second line of thought looked at Sanskrit as a natural language for conversing with computers. This flows from the present developments of knowledge computing systems. There are counter arguments to this aspect also. Many feel that it is difficult to learn Sanskrit and its grammar is more involved. Hence they state that in the present context, Sanskrit may not become a language for communication between users and computers. So one need not be alarmed at the thought that one has to learn Sanskrit to operate a computer or write a program.

Hence it is essential to define the role of Sanskrit clearly with respect to computers. We define initially three views of the relationship between Sanskrit and computers;