An Approach to Sanskrit as Computational and Natural Language Processing

Inderjeet
Department of Computer Science and Applications, Kurukshetra University, Kurukshetra
inderjeetyadav666@gmail.com

Abstract: Sanskrit is a natural language which is comparatively easy to learn with a strong grammer. Many advantages of Sanskrit may find use in some of the frontier areas of computer engineering research, notably in artificially intelligence and knowledge based system. This paper analyses the features of Sanskrit and suggests the use of the same for natural language processing studies and applications. The view point expressed here is that a Sanskrit based compiler or interpreter may have to be developed to unearth the hidden treasures in Sanskrit technical literature. Other country in the west also, of late, have undertaken similar studies and it would only be appropriate if Sanskrit gets the type of recognition that it so richly deserves in its own land even in areas of advanced technological research, for which it is undoubtedly suited. This paper represents how we use an unambiguous natural language for computer programming and for this purpose Astadhayayi plays an important role because it provides a general grammatical framework to analyze languages. This paper describes that how we develop natural language understanding system by the use of Sanskrit grammar.

I. INTRODUCTION

Sanskrit is a natural language and it can serve as an artificial language[1]. Sanskrit includes economics, astrology, science, medicine and mathematics. The simplicity rather than vastness of Sanskrit does make it suitable in many fields. Panini-Backus form or Backus Naur formalism is a grammatical arrangement of words and it is used as natural language processing system, notation for representing the part of natural language grammar, protocols communications and command sets. Naur form has similar

Grammar as advised by Panini so it is called as Panini-Backus Form. Scientists thought that machine translation of one language into another in computer is an easy task but it is very difficult because words can have several meanings. It is possible only by replacing the words in text by their equivalents and modifying and arranging these words according to grammar. Here computational grammar tokens the concept of Karaka and Vibhakti relations and use them to an efficient parse for Sanskrit text.

In Sanskrit Vibhakti helps in making sentences and there are seven types of Vibhakti and on respective Karka Vibhakti provides information. And here sentence contains Subject + verb + predicate. Example – “Goat eats grass”. And if the subject and predicate are interchanged, it is gotten – “Grass eats Goat”. Even sentence is correct but rule of grammatical arrangement of words is ruined because Grass never eats Goat. So making it possible to translate these types of sentences or all sentences there is a requirement of artificial intelligence program to be constructed. Sanskrit shows such power of unambiguity. For example – “Aja Trinam khadati”. This is Subject + Predicate + verb. If the subject and predicate are interchanged, then sentence will be “Trinam Aja khadati”. Meaning remains same as Goat eats grass. And karka helps in generating grammatical relationship of pronouns and nouns. In Sanskrit[2], rules are arranged in several layers and each layer forms an exception for the previous layer.

II. LITERATURE REVIEW

Sanskrit grammar is given by Panini as Astadhyayi[3]. He represents the framework for a universal grammar that may apply to any language. Sanskrit is a spoken language in its time and this is described by Panini. Four thousand aphorisms and rules are contained in his book. Formulas given by Panini for Sanskrit grammar are studied but well explained by Katyayana and Patanjali. Sanskrit is a language of privileged groups like Rishi Munis and Brahmina. A few famous rishi believed that there is a direct relation with God. Example – ‘AUM’ is the famous word. It contains three curves. The large lower curve for working state, one semicircle and dot and upper curve denotes deep sleep and medium curve signify dream state. Its philosophical work is described by Bharthari. To facilitate his description, a technical language is established which indicates rules about rules or Meta rules. To generate words, sentences, rule of transformations Panini grammar[7] termed as Panini machine.

Current grammar of English does not satisfy the conditions. So the researchers would construct a structure which is fit for the capabilities of grammar.

DOI: 10.090592/IJCSC.2015.626

III. DIFFICULITIES IN LANGUAGE TRANSLATION

Machine translation is a difficult task due to different types of ambiguities. To obtain a translation, this different type of ambiguity needs to be resolved.

These difficulties are inherent in English but are not fundamental to all natural languages. Scientific Sanskrit is particularly precise i.e. clear and accurate.

There are different types of ambiguities depending upon word meaning study, problem solving, explanation or understanding and number of meaning of one word etc:

1) **Language translation in structural form-** In this, words in a sentence are interpreted after the sentence combined into groups of words, which are without definite verb. Example – "She saw dragon fly outside". In this dragon fly is long and two winged fly and dragon is a bird which flying. And one another example is "The investigator's arrest was illegal". Here no explanation about who was arrested, the investigator or someone else.
2) **Difficulty or ambiguity in problem solving in practical way-** This is related to sentence context. For example-"Shyam loves his wife and so does Bill" this sentence is unambiguous only if it is known that Bill is a bachelor if there is no description about Bill who is a bachelor. Then this is ambiguous.
3) **Lexical difficulty or ambiguity-** when a single word has many different meanings then this type of difficulty arises and in this all meanings are potentially valid. For example- 'Fan' may refer to object for making a current of air or supporter and lie which may prefer to statement that is not true or put yourself in horizontal position.
4) **Difficulty in study of meaning of words or semantic ambiguity-** This ambiguity is related to sentence. Example – "I like to eat black grapes. This sentence has two meanings – one is that speaker liked a particular bunch of grapes and other meaning is that he expressed his preference for black grapes.

IV. SANSKRIT IS GENERATIVE AND OBJECT ORIENTED

Feature of generating new words is most distinctive feature of Sanskrit language. 14 formulas are given by Panini in Sanskrit language are called 'Siva Sutras'. This explains Sanskrit in mathematical representation or form. Fibonacci series correlated with mathematical expression of language which explains every natural dilemma. Fibonacci series is so simple and nature follows it because it generates the patterns. And this theory explained in Sanskrit as regenerative for computation. Context developing is based on language's grammar. There are 14 basic rules given by Panini and the whole Sanskrit language is a set of all these rules. All the sentences are formed by these 14 basic rules. Classification of all these is possible only through object oriented approaches which is there in Sanskrit grammar. For example- 'likhitwan' indicates that it is a verb in past tense, third person, male and singular in number. In this root "likh" means writing and 'likhitwan' whole meaning is an action of writing in past by single male third person. So Sanskrit has an object oriented approach[8] as compared to any other language[4]. So proper study develops object oriented properties with Sanskrit as base.

Language analysis

In terms of elementary sounds spoken language may be characterized and this characterization is known as phonology. Function in same way of all similar speech sounds without changing in the utterance meaning is called a phoneme. About in 40 phonemes Sanskrit is described in 48 phonemes and in these 48 phonemes each phoneme has a unique symbol in its alphabet. Punctuations, words and alphabets are basic unit of written language. Linguistic analysis is called morphology. Morphemes meaning conveyed by suffixes, plural ending or prefixes.

To form an utterance or sentences meaningful constituents are put together and study of this is known as syntax.

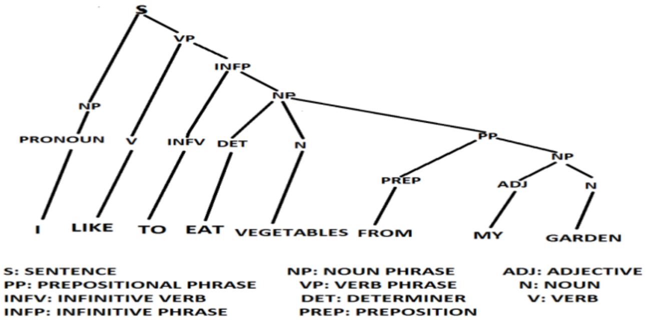

Fig 1 : A Parse Tree

And structure explains in forms of a tree (figure 1 above).
For example – “I like to eat the vegetables from my garden”
And this is represented by a tree-

Method for analyzing Sanskrit text in computer

There are seven karakas which capture several aspects of action through its participants. Panini develops a comprehensive theory for its context relation to its agents and situation and this theory is known as karaka theory.

Different types of karaka-
Karta - agent or one who is independent
Karam- object, what the agent seeks most to attain
Karana- instrument, main cause of the effect
Sampradana – object recipient
Apadana – departure take place
Sambandh – shows relation
Adhikaran- location, basis

And vibhakti provides information on respective karaka. Vibhakti guides for making sentence in Sanskrit.

1. Nominative – prathama vibhakti
2. Accusative – divitya
3. Instrumental – tritiya
4. Dative – chaturthi
5. Ablative - panchmi
6. Possessive – shhashthi
7. Locative – saptami
8. Denominative – astmi

Karaka theory acts as a media between grammar and reality. Karaka do not have one to one correspondence with grammatical case.

Several operation perform in sequence to understand

Several operations were performed on text or speech on computer based system.The first operation is the sound based (Phonological analysis) as well as written based (Morphological analysis). In phonological analysis, sound waves are transcribed in series of phonemes. In morphological analysis each word is decomposed into inflections and roots. Words are according to lexical categories in which one word has more than one meaning. A lexical category includes verb, noun, adjective etc.

The next operation performed is structural analysis which deals with grammar rules used to yield the structure of the sentences. In order to carry out structure analysis first syntactic analysis and then pragmatic analysis is carried out. In syntactic analysis, sentences are converted into a form and in pragmatic analysis, the sentence is made explicit.

After these all analysis computer based system or computer can announce its respond and inferences to question. These operations proceed effectively and meaning of sentences represents in convenient form. There are many operations to understand the system of first Paninian approach for natural language processing.

Mostly effective parsers are based on grammars designed with computer systems in mind. Terry Winograd’s SHRDLU is early system which has limited aims. In this, parsing is done by interpreting grammars written as programs. SHRDLU creates a particular condition i.e. robot that containing box, blocks, cubes and several colored pyramids of different sizes. And this can track and pick up locations and rearrange them. Input system processing initiates and terminated by monitor. Morphemic analysis carried out by input with dictionary and provides grammar.

Parsing of sentences is done by the programmer and answer converts the system response into grammatical language and keeps track of context. For facts about environment and for analysis planner is used. Data contains statements of planner that describes the object being scanned.

Sanskrit analyzing in computer or how machine analyzes or identifies the word in a given sentence as following structure –

Words ⟹ Base ⟹ Forms ⟹ Relations

1. Words – by using the guidelines the parser identifies. Word is broken down into the parts if the given word is compound word. For example - devalaya= dev + alaya
2. Base – the uninflected, and original form of word is base. DFA on the ISCII code activates for Sanskrit text if word is simple. And computer shows nesting of external and internal 'samas' using nested parenthesis.
3. Form – information about the words like action or verbs. Action to be performed is contained in 'Form'. For pronouns and adjectives, write first a, followed by the indication. For nouns, number, gender and case – write 'm', 'f' or 'n' for nouns to indicate the gender, and number for case. And 'p', 'd' or 's' indicates plural, dual or singular. And write 'u' for undeclared words.
4. Relation – gives the relationship between the different words which are used in given sentence.

Sanskrit sutras as base

1. Sutras of paribhasha – it provides Meta language and helps in resolving a conflicts and deadlock conditions and decision are taken by it. The input of our system is 'karaka' level. This is analysis of the nominal stem. The output of our system is the final form which is get after traversing of whole Astadhyayi.
2. Sutras of adhikara – necessary condition for getting triggered (χ) by sutras are assigned by this.
3. Samjna sutra – creation of environment to certain sutras to get triggered done by this.

Algorithm for Sanskrit parser

Sanskrit sentence is taken as input by parser[6] and using the Sanskrit rule base from DFA analyzer, analyze each word and along with their attributes returns the base form of each word. Information is to analyze the relation of words in sentence and the output is a complete dependency parse. We use morphological analyzer because of rich case endings of word of Sanskrit.

The following resources are built for Sanskrit algorithm :

1. Database of particles which contains entries
2. Database of verb rule which contains entries for 10 classes of verbs.
3. Database of nominal rules which contains entries for pronoun and noun declensions.

Morphological analysis (fig 2)

Sanskrit sentences are taken as input in Devanagari format and converted into ISCII format. These words are analyzed using DFA.

Steps for computation –

1. First of all do a left-right parsing[6] for separating the words in sentence.
2. According to DFA, each word is checked against Sanskrit rules base in precedence order.
3. Each word is checked first against database of 'avavya' database, then we check pronoun then verb and at last in the noun tree.

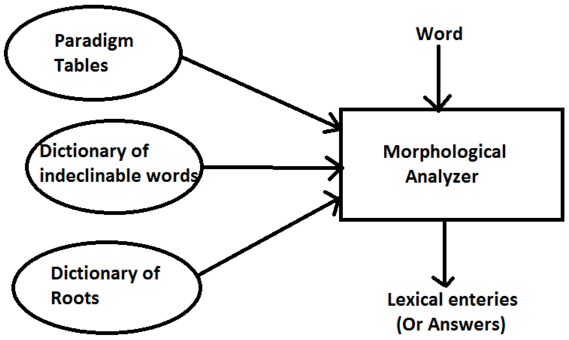
Paradigm Tables
Word
Dictionary of indeclinable words
Morphological Analyzer
Dictionary of Roots
Lexical enteries (Or Answers)

Fig. 2 : Morphological analyzer input-output

Relation analysis

Working in area of processing natural language for extraction of the meaning is challenging in the field of artificial intelligence[1]. In this many Indian and foreign languages are used. But Sanskrit language possesses a definite rule based structure and it has a great potential in the field of semantic extraction. Hence, computational and Sanskrit linguistics[5] are strongly associated. Its case ending is strong identifiers of respective word in sentence.

An algorithm is developed to create a dependency based structure in Sanskrit by analyzing the features of given part of speech. To relate the words with verb in a sentence dependency tags are used. Dependency tags gives the semantic information and part of speech tag gives the information about syntactic.

Sanskrit is an order free language[9]. The Sanskrit language has dependency grammar and this is obtained by using karaka. Karaka helps in generating relation between pronouns and nouns to other words in a sentence. Once the karaka relations are obtained, then it is very easy to get the actual relation of words in the sentence.

V. Future scope

Astadhyayi is a strong grammar formulated with logically consistent rules. These rules completely help to derive correct Sanskrit words and sentences. Due to this Sanskrit resembles as a synthetic language. Not only in the field of linguistics but also in the field of computer science, artificial intelligence and combinatory with respect to compactness, optimality of code, ordering of rules, logical design etc. There are many aspects of Sanskrit grammar rules which are still daunting like the Meta language, concept of Meta rules, priority criteria, dynamic rule hierarchies, partial and extended inheritance etc. It can optimize memory organization and database easily. It can help to remove the ambiguity completely.

VI. CONCLUSION

The first and simple reason to use Sanskrit in computer is- Sanskrit is simple to learn and its grammar is very strong. There is no exception and failure in Sanskrit grammar. Languages people use for their daily communication do not help them in making the distinction required to be in balance with technology. Sanskrit with computer tools will awaken the capacity in human beings to utilize their innate higher mental faculty which keeps the control of the technology within humanity. The researchers do not try to get full semantics immediately but make it appropriate to do so. Here the researchers got encouraging results and they hope to analyze Sanskrit text unambiguously. The researcher’s analysis in the form of relation and morphological analysis for Sanskrit sentences are based on sentences given in paragraphs. The researchers deal with the uncertainty information in Sanskrit grammar of Panini to make it convenient for further computer processing. Vibhakti in Sanskrit enhances the existing work and the would extend the current system and develop a fully fledged parser. Through this NLP (natural language processing) is generated.

REFRENCES

1. Rick Briggs, **knowledge representation in sanskrit and artificial intelligence,** AI Magazine 1985
2. G. Joseph, **Rediscovering Divine Sanskrit,** 2003
3. Katre Sumitra, **Astadhyayi of Panini,** University of texas press, 1989.
4. P.Ramanujum **, A case for Sanskrit as computer programming language**
5. Akshar Bharti and Amba kulkarni, **Sanskrit and computational Linguistics,** 2007
6. Akshar Bharati, Rajiv Sangal and Vineet Chaitanya, **A Karaka based approach to parsing of Indian languages**, 1989
7. Saroja Bhate and Subhash Kak **, Panini’s grammar and computer science** , 1983
8. Willy Smith, **Sanskrit as an object oriented language** , 2005
9. http://www.sanskrit.org

DOI: 10.090592/IJCSC.2015.626