GENERAL STRUCTURE OF SANSKRIT MACHINE TRANSLATION SYSTEM

K. S. Gilda **S. S. Dixit** **S. V. Narote**

Assistant Professor, Department of Computer Science & Engineering,
C.O.E. & T., Akola, Maharashtra, India.

Abstract: Machine Translation (MT) is an emerging field in computer science. It is important for breaking the language barrier and facilitating inter-lingual communication. Though it a fascinating area of research, it faces difficulties right from selecting source-target language pair and designing systems for that language pair. For a multilingual country like INDIA, there is a big requirement of machine translation system. While studying different MT systems in Indian languages it was found that Sanskrit can be thought of source language because of its rich and oldest literature treasure. Many advantages of Sanskrit find use in some of the frontier areas of computer engineering research, notably in artificial intelligence and knowledge based system. The relatively unambiguous nature of this language and its well laid-out grammatical structure promote it as the language for processing.

Success of MT needs issue of Natural Language Understanding (NLU*)* to be solved first. Natural language refers to the language spoken by people such as Hindi, Marathi, Sanskrit, English, etc. as opposed to artificial languages such as C, C++, java etc.

This paper focuses on Sanskrit as source language in Machine Translation system. The paper begins with introduction to MT; then gives features of Sanskrit language and highlights the language features suitable for MT; ends with the discussion of general structure for Sanskrit Machine Translation system (SMTS).

IndexTerms -Machine Translation, vibhakti, karaka, inflection, analyzer.

I. INTRODUCTION TO MACHINE TRANSLATION

Machine Translation (MT) is one of the emerging areas of Artificial Intelligence. MT is subfield of computational linguistics that translates one natural language into another. It involves translation of either text or speech. This paper focuses on text translation only. MT includes understanding the source text and generation of target text. MT is not just word to word translation but meaning of source text should be reflected in target text as if translation is done by human, at the same time target text should be in well-formed according to target language.

This is the point where Natural language processing (NLP) plays a vital role. NLP can be mathematically represented as NLP = NLU + NLG where NLU stands for Natural language understanding and NLG stands for Natural language generation. NLU handles understanding of data, based on grammar, NLG generates text based on structured data and NLP converts text into structured data. NLP involves making computers to perform useful tasks using languages. NLU is a subset of NLP thereby it is narrower in purpose and focuses on machine reading comprehension. NLU involves mapping the given input in natural language into useful representation and analyzing different aspects of the language.NLG involvesText planning i.e. retrieving relevant content from knowledge base and Sentence planning i.e. required word and forming meaningful phrases.NLP has to face a lot of ambiguity during its processing and Sanskrit language overcomes all of these hurdles, because of "formally defined grammar", to become the best suited natural language for MT.

II. SOME SALIENT FEATURES OF SANSKRIT LANGUAGE

In general Language means collection of sentences. A sentence means - collection of meaningfully associated words such as Subject, Object, Verb etc. Each word in a sentence should have clear and easily understandable verbal and nominal declensions. Grammar of Sanskrit is well organized and least ambiguous compared to other natural languages. Sanskrit grammar is given by Panini as "Astadhyayi". Feature of generating new words is most distinctive feature of Sanskrit language. 14 sets are given by Panini in Sanskrit language are called "Maheshwara Sutras", which explain Sanskrit in mathematical representation or form.

These are nothing but rearranged alphabet sets to describe various linguistic properties. For e.g. ह ल् indicates set of all consonants. Panini develops a comprehensive theory for its context relation to its agents and situation and this theory is known as karaka theory. Vibhakti provides information on respective karaka. Vibhakti guides for making sentence in Sanskrit. Karaka theory acts as a media between grammar and reality.

Table1: Maheshwara Sutras [10], [26]

माहेश्वराणि सूत्राणि	
1) अ इ उ ण्	2) ऋ ऌ क्
3) ए ओ ङ्	4) ऐ औ च्
5) ह य व र ट्	6) ल ण्
7) ञ म ङ ण न म्	8) झ भ ञ्
9) घ ढ ध श्	10) ज ब ग ड द श्
11) ख फ छ ठ थ च ट त व्	12) क प य्
13) श ष स र्	14) ह ल्

Following are some salient features of Sanskrit language- [8], [9], [11], [12], [14]

- The relatively *unambiguous nature* of Sanskrit language and its *well laid-out grammatical structure* promote this language as the language for processing.
- Sanskrit has a more *strictly defined syntax*, so it is technically more computable.
- Sanskrit is the most *Scientific and Structured* language. Sanskrit has many hidden Algorithms built into it as part of its vast scientific treatises, for analyzing "Meanings" or "Word sense" from many perspectives.
- The *word representation* in Sanskrit is not done according to the objects instead it is done *by its property*. Any object or a thing is named by the property it possesses.
- All Sanskrit *words* are made of *characters*, either *vowels or consonants*. Vowels exist independently, while consonants depend on vowels. The process of *Sandhi* is defined.
- Sanskrit *words are composed of two parts*, a fixed base part and a variable affix part, both forming an integral unit. The variable part modifies the meaning of the word base, depending on a set of given relationships.
- Sanskrit is a very *predictable language*. It is easy to formulate sentences and get meanings from words. It is *easy to make word's plural*. This means that a computer can inherently formulate sentences very easily.
- Words are of either *nominal type or verbal type* i.e. denoting either entities or actions.
- Only in Sanskrit there is a clear *difference between dual and plural case* and thus we can get an error free NLP.
- *Vibhaktis* (cases) provides an efficient way of *segmenting* the sentences into *logical constructs* for natural language processing (NLP). The splitting of the sentences in Sanskrit is very similar to the semantic net models used for artificial intelligence systems.
- Sentence formation in Sanskrit is done with the help of two well known tools *Vibhakti and Karaka*. Vibhakti guides for making sentence in Sanskrit, there are seven kinds of vibhakti and they also provide information on respective karaka. Karaka approach guides for generating grammatical relationship of nouns and pronouns for other words in a sentence.
- Another unique feature of Sanskrit is its *inflection based syntax* which makes the overall meaning of a sentence independent on the position of its constituent words. An inflection of a word is a different form of that word and is used for enhancing the meaning of the original word.

III. SANSKRIT AND MACHINE TRANSLATION[13]

Sanskrit is one of the very few languages which has formal defined grammar. Its grammar possesses well organized rules and meta rules to infer those rules, thus proving to be a powerful analogy to context free grammar of a computer language. With all its grammatical rules, it can be more or less approximated to a programming language with classes and objects. All the words that are used to make a sentence in Sanskrit are fundamentally properties when appended with a proper case these words can be treated as object.

For any language to become computationally viable, it should possess following features

- Less or Unambiguous Grammar
- Guard against Mispronunciation/ Misspelling Resulting in Misconception
- Total precision
- Co-relation between written and Spoken form of words
- Potential Grammatical Tools

Sanskrit language holds most of these features so it can be treated as best suited natural language for machine translation. The linguistic aspects of Sanskrit language that need to be considered while dealing with complexity in MT are as follows-

Phonetics and Phonology—knowledge about linguistic sounds - In Sanskrit it is known as Panini Shiksha shastra which connects to the Grammar and the rules of the grammar also abide by the rules of the Phonetics.

Morphology—knowledge of the meaningful components of words from stems and their generation and usage - In Sanskrit this is called as 'pada vyutpatti'. In addition the methods for generating words are also explained step-by-step in Panini's Ashtadyayi like a mathematical equation.

Lexical—knowledge of meanings and equivalent words. Every Sanskrit lexical item has a one-one correspondence. So a particular word used in some place means the same when used elsewhere too from a semantics point of view.

Syntax—knowledge of the structural relationships between words - declensions of nominal forms /stems - In Sanskrit Vibhakti play this role – it has very tight rule thus there is no ambiguity.

Semantics—knowledge of meaning of words in a sentence - In Sanskrit this is one discussed in detail in many works and in Sanskrit vyakarana called as "Kaarakam". Many ways of sentence meanings and their analysis on a scientific basis are available in Sanskrit.

Pragmatics— knowledge of the relationship of meaning with respect to the context - this is the most complex as meanings change, based on context and many other factors. In Sanskrit there is a wonderful Vyakarana treatise available for pragmatics called as "Vakyapadiyam" by Maharishi Bhartrhari.

IV. COMPONENTS OF SMTS

Developing a Sanskrit Machine Translation System (SMTS) is much more fascinating and challenging task. MT is 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. The components of proposed Sanskrit Machine Translation system (SMTS) include the modules as shown in figure1.

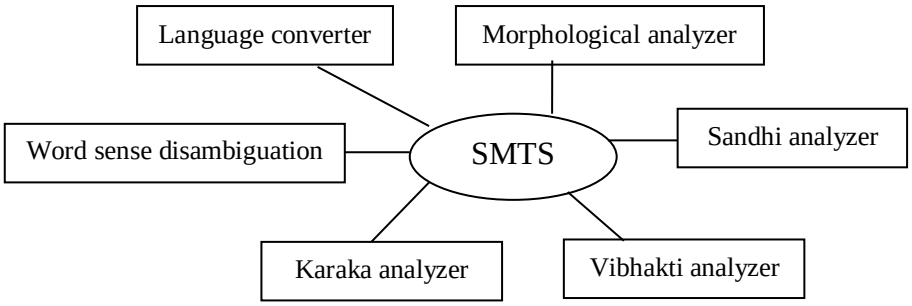

Figure1: SMTS modules.

1. *Morphological analyzer*[1],[2]

We define a word to be a sequence of characters delimited by spaces, punctuation marks etc. It is not difficult to identify words in written text since simply one has to look only for delimiters. A word analyzer is also called as morphological analyzer. Having identified a word, it must determine whether it is simple or compound word. If it is a compound word, it must invoke the next component of SMT system i.e. sandhi analyzer.

2. *Sandhi analyzer*[7],[17], [25], [26]

The compound word must be broken up into constituent simple words before proceeding. When two words combine to produce a new word whose point of combination is result of annihilation of case-end of former word and case-begin of latter, it is called as *Sandhi*. In other words, Sandhi is union of two letters, governed by the rules of Sanskrit grammar.Dissociating the compound words is necessary in order to take out correct meaning. The dissociation of compound words is known as *Sandhi Wicched*.

Sandhi formation is easy task. A set of predefined rules can be applied to form sandhi. There are three types of sandhi-swar-sandhi, vyanjan-sandhi and visarga-sandhi. But, sandhi analyzer is supposed to do Sandhi wicched which is not simple task. From where to break within word, which rule to apply are the issues needed to be handled. Figure shows the example.

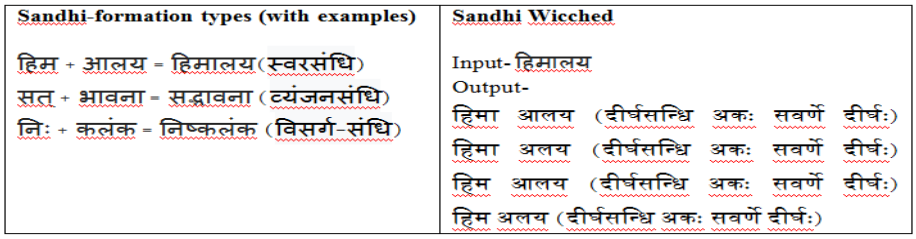

Figure2: Sandhi formation types and Sandhi wicched example.

Another issue is needed to be handled here is analysis of Samaas. When two or more words are combined, based on their semantics then the resulting word is known as *Samaas*. Unlike Sandhi, Samaas may result in elimination of Vibhakti or word. The inverse procedure of break-up of a Samaas is known as *Samaas Vigraha*. In case of Samaas, no such generalized rules are there. Every compound word has different Vigraha as shown in figure.

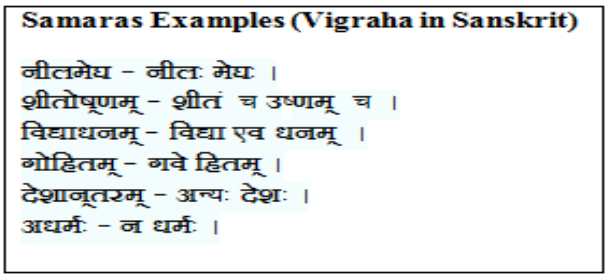

Figure3: Samaas Vigraha examples.

3. *Vibhakti analyzer*[16],[26]

Vibhakti analyzer performs vibhakti identification, which plays very important role in translation process. One of the key tasks in identification of vibhakti is identifying the correct root word from its inflected form. The inflection involves formation of two kinds of words or padas: subanta padas (nominal words) and tianta padas (verb forms). Nominal inflection deals with combination of bases with case suffixes. These extracted suffixes carry sufficient amount of syntactic and semantic information with them. In vibhakti analysis, after the recognition of these padas, the system recognizes all remaining words as subanta and sends for the analysis process. According to Panini, there are 21 morphological suffixes which are attached to the nominal bases according to syntactic category, gender and end character of the base. Sanskrit language has 7 predefined vibhakti (cases) as shown below.

Table2: Vibhakti (cases) in Sanskrit.

Vibhakti	Meaning in English	Example
Nominative	To address nouns (proper/common)	रामः फलं खादति ।
Accusative	The accusative part of the sentence is generally the object on which the action is being performed.	रामः पुस्तकं ददाति ।
Instrumental	It denotes the instrument with which the action is being performed.	रामः लेखन्या लिखति ।
Dative	This case denotes the object for which the object is being performed.	रामः स्वपुत्रेभ्यः कार्यं करोति ।
Ablative	It expresses the point of separation of the object.	रामः स्थात् पतति ।
Genitive	The word declined in the genitive case is the possessor of the object.	रामस्य गृहम् इष्टिकाभिः निर्मितम् ।
Locative	It signifies the location of the object.	रामः भूम्याम् उपविशति ।

4. *Karaka analyzer*[3], [26]

After analyzing individual words it is necessary to understand how the words are arranged in a sentence. A sentence is a group of meaningful words which are inter-related. In the process of forming a sentence, we know that the root –word i.e. verb is most important. Without verb we cannot complete the sentence. That is why other word's importance depends on its relation with verb. This relation of other words in a sentence with verb is called -karaka. There are six types of karaka in Sanskrit language as shown below.

Table3: Karka in Sanskrit.

Karaka	Meaning in English	Example
Karta	agent	दीपः अस्ति।
Karma	patient	दीपं प्रज्वालयति।
Karana	means	दीपेन (चित्रं) दर्शयति।
Sampradana	beneficiary	दीपाय (तैलम्) आनयति।
Apadana	source	दीपात् (प्रकाशं) लभते।
Adhikarana	location	दीपे(वर्तिकां)स्थापयति।

Some potential problems in analyzing karaka are-

- Identifying noun phrases from verb phrases in Sanskrit is a big problem. e.g.
- If verb form in sentence is passive, we need a mechanism to deal with the passive forms.
- For verb-less sentences or clipped sentences, before finding karaka, information provided from the context is needed to be extracted.

5. *Word sense disambiguation* [1], [21]

In computational linguistics, *word-sense disambiguation* is an open challenge of natural language processing. There are a lot of words which denote different meanings in different contexts. Some examples are shown in table. Such words with multiple senses are called ambiguous words and the process of finding the exact sense of an ambiguous word for a particular context is called word-sense disambiguation (WSD). In other words, WSD is identifying which sense of a word (i.e. meaning) is used in a particular context, when the word has multiple meanings. A normal human being has an inborn capability to differentiate the multiple senses of an ambiguous word in a particular context, but the machines need to be programmed accordingly.

Table4: Sanskrit words with dual meaning.

Word	Meanings
राम:	God Ram any common boy named Ram
कर:	Hand Ray
द्विज:	Bird Brahmin
पय:	Water Milk
खद्योत:	Sun firefly / lightning bug

WSD approaches are categorized mainly into three types - knowledge-based, supervised and unsupervised methods. The accuracy of result by particular approach depends on availability of different resources like corpus, tagged data set, WordNet, thesauri etc.

6. *Language converter*

Language converter module uses dictionary (source to target language) to do word by word translation. The words are then needed to be structured according to target language syntax. Case, gender, number and person representation is done according to target language. As these stages are language dependent, so may vary according to target language.

V. CONCLUSION

The specific, unambiguous nature and vast literature and vocabulary of Sanskrit language prompt to be used as source language in machine translation. MT is a difficult task because words can have several meanings. More over the nature, scope and difficulty level of MT depends on source-target language pair. Though Sanskrit has well defined and structured grammar, converting it to other language with different features is a cumbersome task. The significant aspect of our approach is that, we do not try to get the full system immediately; rather it is extracted in stages. Here the main focus of discussion was source language in MT and we proposed a general structure of SMTS which tried to utilize the salient grammatical features of Sanskrit language.

REFERENCES

[1] Akshar Bharati, Vineet Chaitanya and Rajeev Sangal, "Natural Language Processing: Paninian perspective", Prentice Hall of India, 1994.

[2] Aasish Pappu and Ratna Sanyal, "Vaakkriti: Sanskrit Tokenizer", Indian Institute of Information Technology, Allahabad (U.P.), India.

[3] Sudhir Kumar Mishra and Girish Nath Jha, "Sanskrit Karaka Analyzer for Machine Translation", conference paper, 2004.

[4] Manji Bhadra, Surjit Kumar Singh, Sachin Kumar, Subash, Muktanand Agrawal5, R.Chandrasekhar, Sudhir K Mishra, Girish Nath Jha, "Sanskrit Analysis System (SAS)", publication at: https://www.researchgate.net, 2009.

[5] Sanjay Kumar Jha, "A Short Critique of English translations of Sanskrit Terms at lexical level", Veda's Journal Of English Language And Literature (JOELL), Vol.5 Issue 12018.

[6] Pankaj Upadhyay, Umesh Chandra Jaiswal, Kumar Ashish, "TranSish: Translator from Sanskrit to English-A Rule based Machine Translation", International Journal of Current Engineering and Technology, Vol.4, No.5 (Oct 2014).

[7] Sachin Kumar, "Sandhi Splitter and Analyzer For Sanskrit", Phd Thesis, Special Centre for Sanskrit Studies, Jawaharlal Nehru University, New Delhi, India, 2007.

[8] Chandana Bathulapalli, Drumil Desai and Manasi Kanhere, "Use of Sanskrit for natural language processing", International Journal of Sanskrit Research 2016, 2(6): 78-81.

[9] Prajakta R. Chaudhari, Pooja C. Gangurde, Nikhil L. Kulkarni, "Study of Methodologies for utilizing Sanskrit in Computational Linguistics", International Conference on "Emerging Trends in Computer Engineering, Science and Information Technology"- 2015.

[10] Inderjeet, "An approach to Sanskrit as computational and natural language processing", IJCSC, Vol 6 Number 2 April - Sep 2015 pp. 264-268.

[11] Kinjal V Patel, "Sanskrit: Some Insights as a Computer Programming Language", 4th International Conference on Multidisciplinary Research & Practice, 2017.

[12] Shashank Saxena and Raghav Agrawal, "Sanskrit as a Programming Language and Natural Language Processing", Global Journal of Management and Business Studies, Volume 3, Number 10, pp. 1135-1142, 2013.

[13] Vipin Mishra,"Sanskrit as a Programming Language: Possibilities & Difficulties", International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 4, April 2015.

[14] Veda Varidhi P. Ramanujan, "Computer processing of Sanskrit", CDAC, Pune.

[15] K. S. Gilda, S. S. Dixit, "Machine Translation: Approaches and Evaluation," International Journal of Pure and Applied Research in Engineering and Technology, Volume 6 (2), pp. 275-283, September, 2017.

[16] Shweta A Patil and Varunakshi Bhojane, "Vibhakti Indentification technique for Sanskit", International Journal of Engineering Research & Technology (IJERT), Volume 3, Issue 01, pp. 1-6, 2015.

[17] Ravi Pal, Dr. U. C. Jaiswal, "Design & Analysis of an Exhaustive Algorithm for Sandhi Processing In Sanskrit", International Journal of Engineering Research and Development, Volume 4, Issue 8, PP. 33-36, (November 2012).

[18] Saroja Bhate and Subhash Kak, "Panini's Grammar and Computer Science", Annals of the Bhandarkar Oriental Research Institute, vol. 72, pp. 79-94, 1993.

[19] Parul Saxena, Kuldeep Pandey, Vinay Saxena,"Panini's Grammar in Computer Science",Recent Research in Science and Technology, 3(7), pp. 109-111, 2011.

[20] N. Shailaja,"Parser for simple sanskrit sentences based on Paninian Grammar Formalism", PhD thesis, Department of Sanskrit Studies, School of Humanities, University of Hyderabad, June 2009.

[21] Manish Sinha, Mahesh Kumar Reddy .R, Pushpak Bhattacharyya, Prabhakar Pandey, Laxmi Kashyap," Hindi Word Sense Disambiguation", Indian Institute of Technology Bombay, Mumbai.

[22] Jaideepsinh K. Raulji and Jatinderkumar R. Saini, "Sanskrit Machine Translation Systems: A Comparative Analysis", International Journal of Computer Applications (0975 – 8887) Volume 136 – No.1, February 2016.

[23] Akshar Bharati, Sukhada, Prajna Jha, Soma Paul and Dipti M Sharma,"Applying Sanskrit Concepts for Reordering in MT", Language Technology Research Center, International Institute of Information Technology, Hyderabad.

[24] Kevin Knight , Elaine Rich and B. Nair, "Artificial Intelligence", 3E, MGH Education, 2009.

[25] http://susanskrit.org/oral-tradition/2010-07-17-06-20-46.html

[26] Dr.Hansashri Marathe, "Sanskrit for Ayurveda".

Mrs. Kalpana S. Gilda

The author is B.E. (Computer Engineering) from COET, Bambhori, Jalgaon (North Maharashtra University), and ME (CSE) from G.H. Raisoni, SGB Amravati University. She is currently working as Assistant Professor in CSE department in COE&T, Akola (Maharashtra). Her subjects of interest include Artificial Intelligence, Algorithms, Theory of computation, Compiler construction and Operating system. She has keen interest in programming and logic development.

Mrs. Sharwari S. Dixit

The author is B.E.(Electronics), M.Tech (Digital Electronics) from RTM Nagpur University. She is currently working as Assistant Professor in CSE department of COE&T, Akola (Maharashtra). She has keen interest in technical activities and always active in departmental events. Her subjects of interest are Switching theory, Discrete structures, Digital image processing (segmentation) and Digital circuit design.

Ms. Suruchi V. Narote

The author is B.E. (I.T.) from Truba Institute of Engineering & Information Technology,Bhopal, and M.E.CSE from Dr.V. B. Kolte *College* of *Engineering* &Polytechnic, *Malkapur*.She is currently working as Assistant Professor in CSE department in COE&T, Akola (Maharashtra). She also had worked at Tata Consultancy Services as Assistant Systems Engineer (ASE). Her subjects of interest include database management, networking and data structure.