
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2020 SCPE. Volume 21, Issues 3, pp. 543–553, DOI 10.12694:/scpe.v21i3.1783

# A MACHINE TRANSLATION SYSTEM FROM HINDI TO SANSKRIT LANGUAGE USING RULE BASED APPROACH

NEHA BHADWAL*, PRATEEK AGRAWAL†, AND VISHU MADAAN‡

**Abstract.** Machine Translation is an area of Natural Language Processing which can replace the laborious task of manual translation. Sanskrit language is among the ancient Indo-Aryan languages. There are numerous works of art and literature in Sanskrit. It has also been a medium for creating treatise of philosophical work as well as works on logic, astronomy and mathematics. On the other hand, Hindi is the most prominent language of India. Moreover,it is among the most widely spoken languages across the world. This paper is an effort to bridge the language barrier between Hindi and Sanskrit language such that any text in Hindi can be translated to Sanskrit. The technique used for achieving the aforesaid objective is rule-based machine translation. The salient linguistic features of the two languages are used to perform the translation. The results are produced in the form of two confusion matrices wherein a total of 50 random sentences and 100 tokens (Hindi words or phrases) were taken for system evaluation. The semantic evaluation of 100 tokens produce an accuracy of 94% while the pragmatic analysis of 50 sentences produce an accuracy of around 86%. Hence, the proposed system can be used to understand the whole translation process and can further be employed as a tool for learning as well as teaching. Further, this application can be embedded in local communication based assisting Internet of Things (IoT) devices like Alexa or Google Assistant.

**Key words:** Rule based approach, Natural Language Translation, Parts of speech tagging, Sanskrit Translation, Hindi Translation

**AMS subject classifications.** 68T50

## 1. Introduction.

Machine translation is defined as the branch of artificial intelligence which covers the task of converting a source language to any other target language from the set of natural languages. The meaning of the text should be preserved and output should be fluent as well as correct. It is one of the applications of natural language processing (NLP), which is the study of interaction between human languages and the computers. It is important for the computers to read, analyze, understand and derive meaningful information from the human natural language. This needs to be done in an accurate and optimized way. Apart from machine translation, other applications of NLP include sentiment analysis, speech recognition, auto summarizing and topic segmentation.

### 1.1. NLP phases.

Major phases involved in NLP, once the system receives the input, are as follows:

#### 1.1.1. Morphological processing.

It is the study of recognizing how a base word is modified to form words having similar meanings and syntactical structures.

#### 1.1.2. Lexical analysis.

It is the process of dividing the whole text into smaller units called lexicons. The lexicon of a language can be a paragraph, a sentence or a word.

#### 1.1.3. Syntactical analysis (Parsing).

It is the process which analyses the arrangement of the words which shows their relationship and checks the sentence for grammar.

#### 1.1.4. Semantic analysis.

It is the process of extracting and checking the dictionary meaning of each word. The text is checked for meaningfulness.

*School of Computer Science & Engineering, Lovely Professional University, Phagwara, Punjab, India (bhadwalneha21@gmail.com)

†School of Computer Science & Engineering, Lovely Professional University, Phagwara, Punjab, India and Institute of ITEC, Univeristy of Klagenfurt, Austria (prateek061186@gmail.com) : corresponding author

‡Department of Computer Science & Engineering, Lovely Professional University, Phagwara, Punjab (India) (vishumadaan123@gmail.com)

Table 1.1
*A Brief Comparison of Hindi and Sanskrit Grammar*

| **Basis** | **Hindi** | **Sanskrit** |
|---|---|---|
| **Alphabets** | 45 characters (varnas) | 46 characters (varnas) |
| **Total Vowels** | 12 vowels (swaras) | 13 vowels (swaras) |
| **Total Consonants** | 33 consonants (vyanjanas) | 33 consonants (vyanjanas) |
| **Number** | 2; singular plural | 3; singular, dual, plural |
| **Gender** | 2; masculine, feminine | 3; masculine, feminine, neuter |
| **Person** | 3; first, second, third | 3; first, second, third |
| **Ordering of Sentence** | S-V-O (Subject-Verb-Object) | Order free language |
| **Total Tenses** | 3; Present, Past, Future | 6; Aorist, Perfect, Present, Imperfect, First future, Second future |
| **Verb Mood** | 3; Indicative, Imperative, Subjective | 4; Imperative, Conditional Benedictive, Potential |

Table 1.2
*Vibhakti-Kaaraka relationship*

| **Vibhakti** | **Karaka** | **Meaning** |
|---|---|---|
| **Nominative** | Kartaa | Performer/ Subject |
| **Accusative** | Karama | Object |
| **Instrumental** | Karana | Instrument |
| **Dative** | Sampradana | For whom the action is performed |
| **Ablative** | Apadana | From where (place) the action is performed |
| **Genitive** | Sambandha | Denotes possession |
| **Locative** | Adhikarana | Location |
| **Vocative** | Sambodhana | Used to address someone |

**1.1.5. Discourse integration.** It is the study of the relationship between any two sentences in a text. The meaning of one sentence may depend upon the preceding sentence and also brings about the meaning of next sentence.

**1.1.6. Pragmatic analysis.** It is the process which derives such aspects of the natural language that require real world knowledge. The text is re-interpreted to convey what it actually meant.

The benefits of using a machine translator instead of doing it manually are numerous. It is time-saving and optimizes effort and cost. It can also provide confidentiality and multi-lingual support which may not be possible in case of manual translation. However, problems may arise if context consideration is not taken into account. Similarly, languages may be ambiguous or have different structures. Also, machine translation is still not fully accurate.

**1.2. Linguistic Features of Sanskrit.** The Sanskrit language is one of the ancient Indo-Aryan languages. Vedic Sanskrit has been used to write many of the ancient documents. The classical Sanskrit is described in a famous grammar called Astadhyayi (Eight Chapters) composed by Panini. Sanskrit is written in Devanagari script as well as in various regional scripts. There are numerous works of drama and poetry in Sanskrit. It has also been a medium for creating treatise of philosophical work as well as works on logic, astronomy and mathematics. Grammatically, it is similar to Indo-European languages as it is an inflected language. Sanskrit language comprises of 46 characters (varnas) out of those, there are 33 consonants (vyanjanas) and 13 vowels (swaras). Sanskrit language comprises of a total of eight cases. There are six tenses (kaala). In addition, there are four moods (arthaa). These four moods and six tenses are together known as a total of ten Lakaaras of the Sanskrit grammar. In addition, Sanskrit word order is free which means most of the sentences can be read and written in free-word-order.

**1.3. Comparison of Hindi and Sanskrit language.** Hindi is one of the official languages of India [5]. It is the fourth most widely spoken language in the world after Mandarin, English and Spanish [22]. Hindi is directly derived from Sanskrit language. It is regarded as the Apabhramsha (corrupted version) of Prakrit which is the Apabhramsha of Sanskrit. Though both the languages have same root, there are grammatical differences between Hindi and Sanskrit.

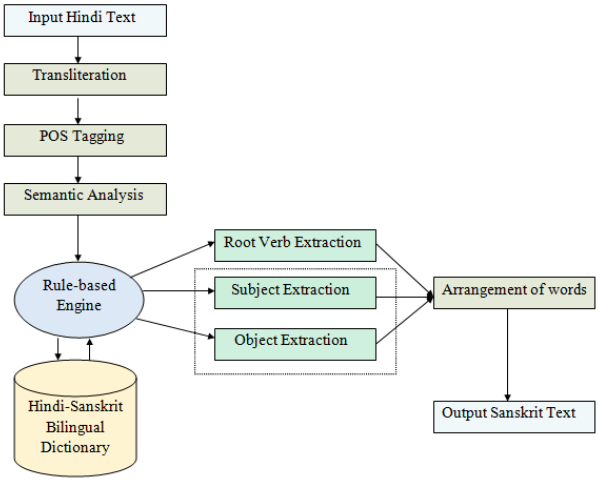


FIG. 1.1. *Architectural Work-flow of Hindi to Sanskrit Translation System (HST)*

Hindi has two types of vowels, short (Hrasva svaras) and long (Deergha svaras) vowels while Sanskrit has three types of vowels, short (Hrasva svaras), long (Deergha svaras) and elongated (Pluta svaras) vowels. In Sanskrit, the last syllable of the word is pronounced completely unless it is marked with a halanta (symbol). For Hindi, word order matters while for Sanskrit it does not matter. There are three basic parts of speech in Sanskrit namely, Shabda (nouns, pronouns, adjectives), dhaatu (verbs) and avyaya (indeclinables). Table 1.1 presents a comparison of Hindi and Sanskrit grammar.

**1.3.1. Nouns and Gender.** In Sanskrit, every noun has 24 forms which is a combination of a case (vibhakti) and a number (vachana). Also, each noun has a gender. However, adjectives in Sanskrit do not have fixed gender. It should comply with the noun it describes in gender, case and number.

**1.3.2. Vibhaktis.** A word can have eight possible vibahktis. Six of these, which are related to actions, are called kaarakas. Possessive and denominative are not related to any action. Table 1.2 depicts the association between vibhakti and kaaraka.

**1.3.3. Sandhi.** When a certain set of letters come together either between or within words, there occurs certain euphonic changes. Depending on the type of letters involved in coalescence, Sandhi can be categorized as Swara Sandhi, Vyanjana Sandhi and Visarga Sandhi.

The paper is organized as follows. The main results are in 4, the proposed algorithm is in 3, and the conclusions follow in 5. The previous related work is in 2.

**2. Previous Work.** A method is presented in [14] to perform syntax analysis of Hindi sentences via a parsing technique based on probability. This technique makes use of CYK (Cocke–Younger–Kasami) parsing algorithm. This morphological analyzer tool was used to identify whether the Hindi sentence is right or not when studied semantically by creating parse tables and making use of morphological information with around 80% accuracy in the syntax analysis stage around 89% in the semantic analysis phase (given the syntax analysis phase generates positive results). A rule-based lexicon parser is proposed by [23] for parsing tokens of Hindi sentence. The Hindi sentence was analyzed both syntactically and semantically after it was tokenized. This analysis was performed with the help of Hindi Wordnet by applying multiple tags. This parser was claimed to produce an accuracy of around 89% when evaluated for different sentences. English-Sanskrit-Hindi translation pair divergences are presented by [8]. The authors study English-Sanskrit and Hindi Sanskrit language pairs to identify divergences specific to each language involved. Verb frames for Hindi language were developed by [4]. A linguistic tool was developed to analyze and understand Hindi verbs. A classification of verbs was performed

**Algorithm 2** HST algorithm

**procedure** HST(W, $V_w$, P, C, F, T, G, N, $P_v, S_v, P_n, S_n, P_1, P_2, D$)
  **W**: Set of all Hindi words given in the sentence
  **Vw**: Set of all verb form words in Hindi
  **P**: Set of all prepositions in Hindi
  **C**: Set of all cases in Hindi
  **F**: Set of all noun word forms in Hindi
  **T**: Set of all tenses in Sanskrit
  **G**: Set of all genders in Sanskrit
  **N**: Set of all numbers in Sanskrit
  **Pv**: Set of prefixes for Sanskrit verb form words
  **Sv**: Set of suffixes for Sanskrit verb form words
  **Pn**: Set of prefixes for Sanskrit noun form words
  **Sn**: Set of suffixes for Sanskrit noun form words
  **P1**: Set of translated Sanskrit noun words
  **P2**: Set of translated Sanskrit verb words
  **D**: Set of Sanskrit words in sentence after translation
  Set $S_{in}, S_{out}$ as empty
  **FOR EACH** $w_i$ in W
    **IF**$w_{i+1}$ ! = **NULL AND** $w_{i+1}$ = = ' है' | ' था' **THEN**
      **IF** $w_i$.contains(' कर' | ' बन' | ' लग') **THEN**
        $S_{in} \leftarrow (w_{i-1}$ U $w_i$ U $w_{i+1})$
      **ELSE**
        $S_{in} \leftarrow (w_i$ U $w_{i+1})$
      **ENDIF**
      Compare reverse [$S_{in}$] with predefined ending phrase list E[n] from corpus
      Store tense 't', number 'r' and gender 'g' corresponding to E[n]
      **IF** substrings $s_1, s_2, s_3 \ldots s_m \in S_{in}$match with E[n] **THEN**
        $S_{out} \leftarrow MAX$ [length ($s_1$), length ($s_2$)...length($s_m$)]
      **ENDIF**
      $S_v \leftarrow (S_{in} - S_{out})$
      **IF** $S_v == V_w$ **THEN**
        Store Sanskrit verb, $V_s$ corresponding to $V_w$ present in corpus
      **ENDIF**
      **IF** (t & r & g) = = (T[n] & N[n] & G[n]) **THEN**
        Store prefix, $P_v$ and suffix, $S_v$ corresponding to (T[n] & N[n] & G[n]) from corpus
      **ENDIF**
    **ENDIF**
    $P_2 \leftarrow P_2 \cup (P_v \cup V_s \cup S_v)$
  **ENDFOR**
  **FOR EACH** $w_i$ in (W-$S_{in}$)
    **IF** $w_{i+1}$ ! = **NULL AND** $w_{i+2}$ ! = **NULL AND** $w_{i+1}$ = = (' और' | ' या' | ' अथवा') **THEN**
      $w_i \leftarrow (w_i \cup$ ' ने')
      $w_{i+2} \leftarrow (w_{i+2} \cup$ ' ने')
    **ELSE**
      $w_i \leftarrow (w_i \cup$ ' ने')
      $w_{i+2} \leftarrow (w_{i+2} \cup$ ' को')
    **ENDIF**
    **IF** $w_{i+1}$ ! = **NULL AND** $w_{i+1}$ = = P **THEN**
      **IF** $w_{i+1}$ = = ' ने' **THEN**
        $c \leftarrow nominative$
      **ELSE IF** $w_{i+1}$ = = ' को' **THEN**
        $c \leftarrow accusative$
      **ELSE IF** $w_{i+1}$ = = ' के लिए' **THEN**
        $c \leftarrow dative$
      **ELSE IF** $w_{i+1}$ = = (' का' | ' के ' | ' की')**THEN**
        $c \leftarrow genitive$
      **ELSE IF** $w_{i+1}$ = = (' में' | ' पर') **THEN**
        $c \leftarrow locative$
      **ELSE IF** $w_{i+1}$ = = ' से' **THEN**
        **IF** $w_{i+2}$.contains (' कर') **THEN**
          $c \leftarrow ablative$
          i = i+1
        **ELSE**
          $c \leftarrow instrumental$
        **ENDIF**
        i = i+2
      **ENDIF**
      Store word form $w_f$ as last character of $w_i$
      $w_f \leftarrow$ last [$w_i$]
      **IF** (c & $w_f$ & r) = = (C[n] & F[n] & N[n]) **THEN**
        Store prefix $P_N$ and suffix $S_N$ corresponding to (C[n] & F[n] & N[n]) from the corpus
      **ENDIF**
      $P_1 \leftarrow (P_1 \cup P_N \cup w_i \cup S_N)$
  **ENDFOR**
  Combine two outputs to give final output D
  $D \leftarrow (P_1 \cup P_2)$
  Display D

and verb frames were created using the kaaraka relationships of the verbs. The criteria of classification was the syntactic and semantic differences between the verbs.

Natural language processing technique is applied by [11] to parse Hindi words and to extract the root words after performing stemming on each individual word. A tool is implemented by [26] to paraphrase the Hindi sentences by using active-passive voice rules and synonym-antonym replacement methods. An algorithm for the transliteration from English to Sanskrit text is presented providing 100% accuracy [12]. The process performs the mapping by making use of Hindi Unicode characters.

Also, a review of various types of MTS is presented by [24]. In their survey, the researchers highlighted and categorized mainly used approached for machine translation system as; rule-based, corpus-based and hybrid machine translation. In addition, [18] provided a view of example based machine translation system (EBMT). It was then compared with RBMT and SMT systems. The prominent characteristics of Sanskrit grammar and a comparison of English language and Sanskrit language were presented. The paper has shown the divergence between English and Sanskrit language with the help of illustrative examples. An overview of a FOS (Free and Open Source) rule-based MTS named Apertium is given by [7].A comparison of rule-based MT and Statistical MT was performed keeping in account the origin of the languages [29]. A five-way comparison between RBMT and SMT followed by English and Indian origin language was done.

An evaluation method is given by [15] which they applied to the three major MT approaches namely, rule-base, phrase-based and neural MT. They have used a case study as the basis of their research work, which performs the translation from English to German language. A number of research efforts were reviewed by [28] under the example-based machine translation paradigm and attempted to categorize them into various classes. The features of various EBMT systems were discussed. The limitations and advantages of an EBMT system were highlighted.

Statistical methods were discussed by [6] along with pre-processing and post-processing of words to improve the performance of English to Arabic language MT such as morphological tokenization, syntactic reordering, orthographic normalization, morphological de-tokenization, orthographic de-tokenization and orthographic enrichment etc. An MTS was designed by [10] for English to Finnish translation, which uses rule-based approach. An empirical study was performed by [21] of the top five state-of-the-art machine translation systems from English to the languages which were considered to be having lesser resources namely, Lao(la), Myanmar(mm) and Thai(th) in both the directions. Various methods of MT were emphasised such as string-to-tree, tree-to-string, phrase-based, hierarchical phrase-based, operational sequence model statistical MTS.

A machine translation system based on rule-based approach was proposed by [17], which worked from English to Sanskrit language. The proposed approach, makes use of a set of transfer rules which are hand-written and convert the lexicons (could be paragraphs, sentences or phrases) from English to Sanskrit language. On a similar note, a process engine named EtranS which accepts sentence in English and produces its equivalent sentence in Sanskrit after translation, was developed by [3]. The approach followed was rule-based and they discussed firstly some syntactical features of the Sanskrit language followed by a comparison between Sanskrit grammar and Context Free Grammar. Another rule-based approach to manifest knowledge representation of a machine translation procedure from Sanskrit to English language was presented by [9]. Various MTS which involve Sanskrit as either a source, target or a key support language are discussed by [13]. They also presented different techniques used by researchers for machine translation, such as, Corpus based, Rule based and Direct translation. The principal objective of this paper was to find out the Sanskrit language suitability, morphology and apply most suitable MT techniques. Furthermore, an RDR POS tagger was developed by [20] to tag the words based on part of speech of the given sentence. A rule-based MT approach from Hindi to English was proposed by [16]. In his Ph.D. thesis report, [1] proposed a machine translation system for Sanskrit to Hindi language.

[2] provide an overview of Machine Learning giving an insight to why it is the future of computing industry. Ant Colony Optimization algorithm and its variants are reviewed based on various categories of problems they are applied to by [27]. [19] highlights various state-of-the-art Virtual Reality (VR) & Augmented Reality (AR) technologies that will prove to be beneficial for tourism and hospitality industry. A cumulative analysis of multiple text document classification algorithms has been discussed by [25].

TABLE 2.1
*Steps followed by HTS*

| Step | Description | Output |
|---|---|---|
| 1 | Input text provided to the system | raama raavaNa ko baaNa se dharma kee rakshA keliye ayodhyaa se jaakara maarataa hai \| |
| 2 | Transliterat- ion of the input text | राम रावण को बाण से धर्म की रक्षा के–लिये अयोध्या से जाकर मारता है \| |
| 3 | POS tagging of input text | राम /NNPC रावण/NNP को/PSP बाण/NN से/PSP धर्म/NN की/PSP रक्षा/NNPC के लिये/NN अयोध्या/NNP से/PSP जाकर/VM मारता/NN है/VAUX |
| 4 | Semantic analysis of input text | राम ने रावण को बाण से धर्म की रक्षा के लिये अयोध्या से जाकर मारता है \| |
| 5 | Explanation of the analysis | राम / पुल्लिन्ग् / १<br>ने / संबंध सूचक अव्यय<br>रावण / पुल्लिन्ग् / २<br>को / संबंध सूचक अव्यय<br>बाण / बाण / ३<br>से / संबंध सूचक अव्यय<br>धर्म / पुल्लिन्ग् / ६<br>की / संबंध सूचक अव्यय<br>रक्षा / स्त्रिलिन्ग् / ४<br>के लिये / संबंध सूचक अव्यय<br>मारता है / पुल्लिन्ग्<br>लट्लकार<br>एकवचन् / प्रथम |
| 6 | Final translation | रामः रावणम् बाणेन् धर्मस्य रक्षायै अयोध्यायाः गत्वा हन्ति \| |

TABLE 2.2
*Sample for Semantic Analysis of tokens (Hindi words & phrases)*

| S.N. | Input | Expected Output | Actual Output | Match |
|---|---|---|---|---|
| 1 | राम | रामः | रामः | Yes |
| 2 | जाता है | गच्छति | गच्छति | Yes |
| 3 | और | च | च | Yes |
| 4 | पढ़ते हैं | पठतः | पठतः | Yes |
| 5 | विद्यालय | विद्यालयम् | विद्यालयम् | Yes |
| 6 | राम को | रामम् | रामः | No |
| 7 | खेलना चाहिए | क्रीडेत् | क्रीडेत् | Yes |
| 8 | पूजा करते हैं | अर्चतः | अर्चतः | Yes |
| 9 | खा रहा था | अखादत | अखादत | Yes |
| 10 | जाएगा | गमिष्यति | गमिष्यति | Yes |
| 11 | देखते हैं | पश्यन्ति | पश्यन्ति | Yes |
| 12 | हम सब | वयम् | वयम् | Yes |
| 13 | देते हो | यच्छसि | यच्छसि | Yes |
| 14 | सीता | सीता | सीता | Yes |
| 15 | रावण को | रावणम् | रावणम् | Yes |
| 16 | रहना चाहिए | वसेयम् | वसेयम् | Yes |
| 17 | बहती है | प्रवहति | प्रवहति | Yes |
| 18 | नदी पर | नद्याम् | नदीम् | No |
| 19 | मारे | हन्तु | हन्तु | Yes |
| 20 | बैठती हैं | तिष्ठन्ति | तिष्ठन्ति | Yes |

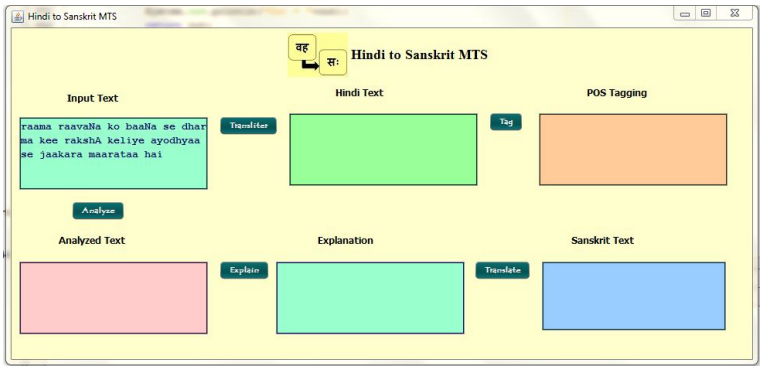


FIG. 2.1. *Providing input text to the system*

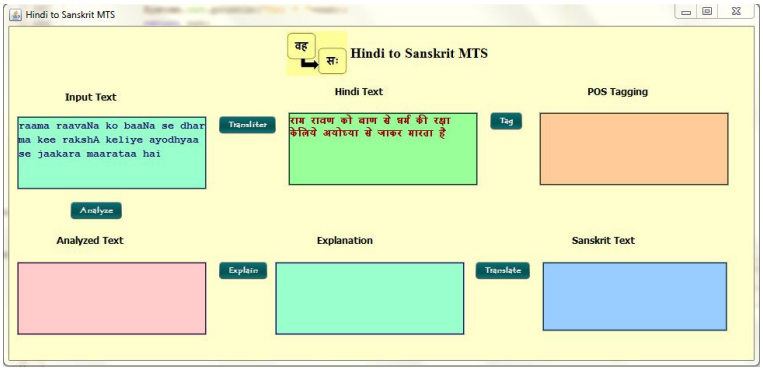


FIG. 2.2. *Transliteration of input text*

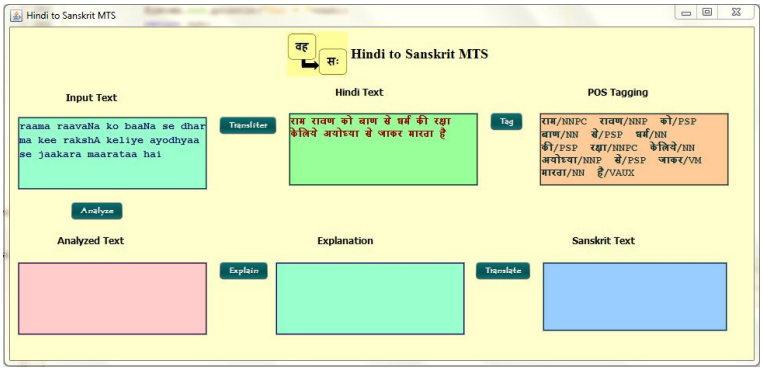


FIG. 2.3. *POS tagging of input text*

TABLE 2.3
*Confusion matrix for Semantic Analysis*

| | | **Output tokens** | | |
|---|---|---|---|---|
| | | **C** | **I** | **Total** |
| | **C** | 74 | 06 | 80 |
| **Input tokens** | **I** | 00 | 20 | 20 |
| | **Total** | 74 | 26 | 100 |

*** C** - Correct **I** - Incorrect

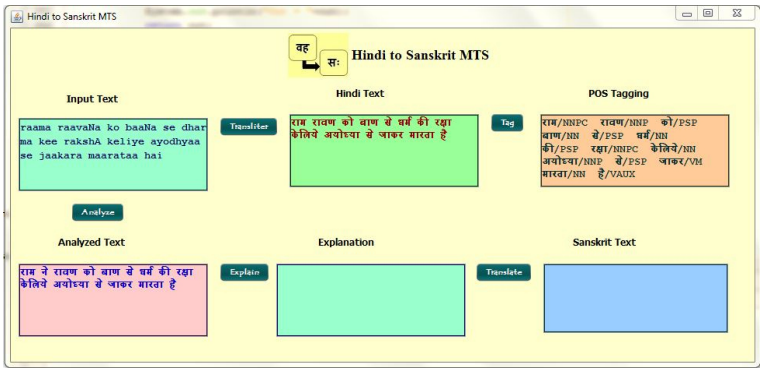


FIG. 2.4. *Semantic analysis of input text*

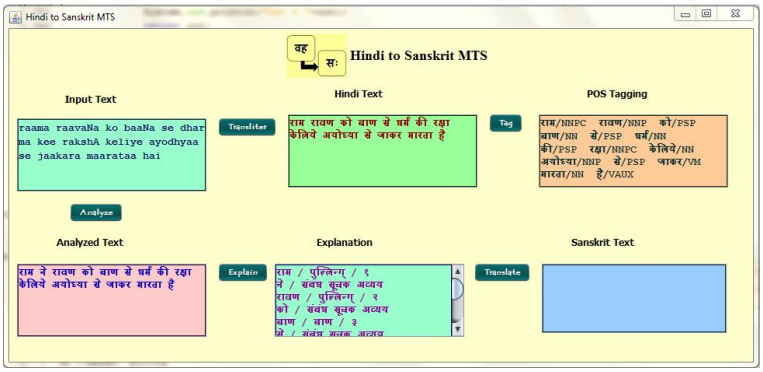


FIG. 2.5. *Explanation of the analysis*

**3. Proposed Methodology.** The Hindi to Sanskrit Translation system (HST) is a rule-based model. The prototype model that we have developed is a machine translation system from Hindi to Sanskrit language and it makes use of the rule-based approach. A rule-based machine translation system (RBMT) makes use of dictionaries (bilingual or multilingual) and grammars of both the source and target languages. This linguistic information contains the semantic, syntactical and morphological patterns of each of the languages. An RBMT system can either be a dictionary based MT, a transfer based MT or an Interlingua. Our system is a direct system (dictionary based MT) which maps input (Hindi text) to output (Sanskrit text) with basic rules. We have developed a Java application for implementation purposes. The system is fast and easy to implement. Moreover, the accuracy and performance can be further improved by the addition of more complex rules to cover the overall features of the language.

The proposed model takes in as input a Hindi text, processes it and produces the corresponding Sanskrit text as output as shown in figure 2.6. The processing phase is divided into multiple modules. These modules are described in the following subsections and algorithm 2 presents the steps involved:

**3.1. Transliteration module.** Transliteration refers to the process of converting a set of characters that are in the source language to a set of characters that are in the target language which have similar pronunciation. In other words, the phonetic similarity of the two languages is taken into account for transliteration. We have used a set of rules to perform this mapping from Latin to Devanagari script giving a 100% accuracy based on [12].

**3.2. POS tagging module.** POS (Part-Of-Speech) tagging is referred to as the task of assigning every word in a text to a particular part of speech such as, noun, pronoun, verb, adjective, adverb, etc. This tagging depends upon the actual meaning and the context of the word in the text. We have used the RDR (Ripple Down Rule based) POS tagger which uses an error-driven approach to construct tagging rules in the form of

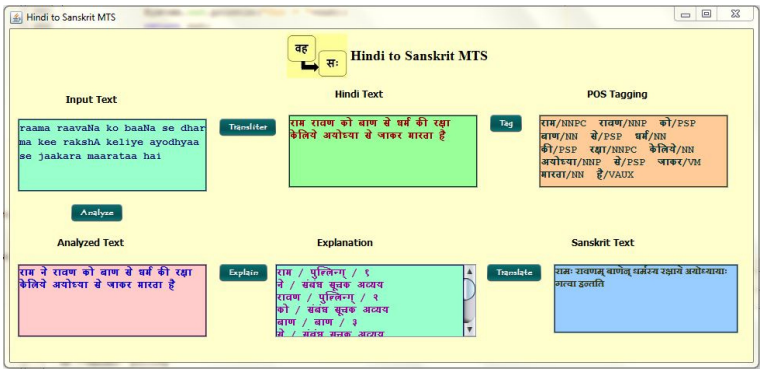


FIG. 2.6. *Final translation*

TABLE 2.4
*Sample for Pragmatic Analysis of Hindi sentences*

| Sentence | Actual Input | Ideal Input | Expected Translation | Actual Translation | Match |
|---|---|---|---|---|---|
| S-1 | सीता पुस्तक पढ़ती है | सीता पुस्तक पढ़ती है | सीता पुस्तकम् पठति | सीता पुस्तकम् पठति | Yes |
| S-2 | राम को खेलना चाहिए | राम को खेलना चाहिए | रामः क्रीडेत् | रामः क्रीडेत् | Yes |
| S-3 | राम और श्याम पूजा करते हैं | राम और श्याम पूजा करते हैं | रामः श्यामः च अर्चतः | रामः श्यामः च अर्चतः | Yes |
| S-4 | राम रावण को मारे | राम रावण को मारे | रामः रावणम् हन्ततु | रामः रावणम् हन्ततु | Yes |
| S-5 | गीता नदी पर जाती है | गीता नदी पर जाती है | गीता नद्याम् गच्छति | गीता नदीम् गच्छति | No |
| S-6 | राम फल खा रहा था | राम फल खा रहा था | रामः फलम् अखादत | रामः फलम् अखादत | Yes |
| S-7 | राम विद्यालय जाएगा | राम विद्यालय जाएगा | रामः विद्यालयम् गमिष्यति | रामः गमिष्यति | Yes |
| S-8 | राम, श्याम और सीता विद्यालय जाते हैं | राम, श्याम और सीता विद्यालय जाते हैं | रामः, श्यामः सीता च विद्यालयम् गच्छन्ति | रामः, श्यामः सीता च विद्यालयम् गच्छन्ति | Yes |
| S-9* | तुम दोनो जाना चाहिए | तुम दोनो को जाना चाहिए | युवाम् गच्छेतम् | युवाम् गच्छेतम् | Yes |
| S-10* | राम और श्याम जाता है | राम और श्याम जाते हैं | रामः श्यामः च गच्छतः | रामः गच्छति | No |

* - Semantically incorrect sentence

a binary tree [20]. The tagger produces an accuracy of 95.77 for Hindi language which is a key factor for our translation system.

**3.3. Rule-base engine.** We have created a database to store rules for translation from Hindi to Sanskrit text. The database consists of multiple rules which are being used to identify and map verbs, nouns, etc. This is done by identifying the number (vachana), gender, case and person (purusha). The conversion from source to target language takes place by comparing, mapping and identifying the corresponding values from the corpus.

**3.4. Root Verb extraction module.** This module outputs the equivalent Sanskrit verb for the input Hindi verb. The root verb depends upon person, gender and number of the noun to which the verb corresponds to. The algorithm first tries to extract the person, number and case for the verb. Then, it extracts the root verb, compares and finds an appropriate mapping from Hindi to Sanskrit.

TABLE 2.5
*Confusion matrix for Pragmatic Analysis*

| | | **Output tokens** | | |
|---|---|---|---|---|
| | | **C** | **I** | **Total** |
| | **C** | 30 | 05 | 35 |
| **Input tokens** | **I** | 02 | 13 | 15 |
| | **Total** | 32 | 18 | 50 |

*** C** - Correct **I** - Incorrect

**4. Implementation and Results.** We have implemented our Hindi to Sanskrit translation (HST) system in Windows using Java. A GUI (Graphical User Interface) has been developed wherein the user is prompted to enter the Hindi sentence as input from the keyboard (in Latin script) and the end result, in the form of Sanskrit sentence, is displayed on the interface. The input text is first transliterated to phonetically similar Devanagari script, Ripple Down Rule-based Parts Of Speech (RDR-POS) tagging is performed to identify different parts of speech, input sentence is then analyzed semantically and finally, the translation to Sanskrit is performed. It also shows the result after transliteration and POS tagging (using RDR POS tagger) of the input Hindi sentence before translation to Sanskrit. Various steps followed by the system are explained and shown in figure 2.1, 2.2, 2.3, 2.4, 2.5, and 2.6. The description and output of every step is presented concisely in table 2.1.

Semantic analysis is applied to tokens which eventually combine to form sentences. Out of 100 tokens, 80 are correct for which a sample is displayed in the form of table 2.2. The 20 incorrect tokens are not recognized by the system. Since no output is produced for them they are not displayed in the table. Table 2.3 depicts the confusion matrix created for the analysis purpose.

Accuracy = (74+20)/100 = 94/100 = 0.940

Error Rate = (6+0)/100 = 6/100 = 0.060

Pragmatic analysis is performed on 50 sentences, out of which 35 are correct and 15 are incorrect. The sample for analysis is displayed in table 2.4. Table 2.5 depicts the confusion matrix created for the analysis purpose.

Accuracy = (30+13)/50 = 43/50 = 0.860

Error Rate = (5+2)/50 = 7/50 = 0.140

**5. Conclusions and Future Work.** The work proposed above strives to translate different kinds of possible Hindi sentences to equivalent Sanskrit sentences. All of these modules provide a deep and thorough understanding of the interaction between the two languages and their translation process. The rule-based approach can explain the detailed comparison of the two languages of interest and the logical process of how we reach a particular result. These features make this translation system an excellent tool for study involving languages and their interaction. Apart from being a learning tool, it can be used as a teaching pedagogy tool. This will be beneficial for the students and teachers who are involved in the process of learning and teaching Sanskrit language.

In future, we can try to translate interrogative and more complex type of sentences. We can also attempt to use other approaches to machine translation apart from the rule based approach and compare the results. The system can be enhanced to support voice translation taking speech (in source language) as input and producing speech (in target language) as output. We can try to minimize the processing time and the memory requirements of the translator so that the use of the computer resources is optimized.

REFERENCES

[1] P. AGRAWAL, *A Machine Translation System for Sanskrit to Hindi language*, PhD thesis, IKG-Punjab Technical University, 2018.

[2] J. ALZUBI, A. NAYYAR, AND A. KUMAR, *Machine learning from theory to algorithms: An overview*, Journal of Physics: Conference Series, 1142 (2018), p. 012012, https://doi.org/10.1088/1742-6596/1142/1/012012.

[3] P. BAHADUR, A. JAIN, AND D. S. CHAUHAN, *Architecture of english to sanskrit machine translation*, 2015 SAI Intelligent Systems Conference (IntelliSys), (2015), https://doi.org/10.1109/intellisys.2015.7361204.

[4] R. BEGUM, S. HUSAIN, L. BAI, AND D. M. SHARMA, *Developing verb frames for hindi*, in LREC, 2008.

[5] G. o. I. Department of Official Language, *Constitutional Provisions: Official Language Related Part-17 of The Constitution Of India*, 2015.
[6] S. Ebrahim, D. Hegazy, M. G. M. Mostafa, and S. R. El-Beltagy, *English-arabic statistical machine translation: State of the art*, Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science, (2015), p. 520–533, https://doi.org/10.1007/978-3-319-18111-0_39.
[7] M.L. Forcada, M. Ginesti-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J.A. Pérez-Ortiz, F. Sánchez-Martinez, G. Ramí rez-Sá nchez, F.M. Tyers, *Apertium: a free/open-source platform for rule-based machine translation*, Machine Translation 25 (2) (2011), p. 127–144, https://doi.org/10.1007/s10590-011-9090-0.
[8] P. Goyal and R. M. K. Sinha, *Translation divergence in english-sanskrit-hindi language pairs*, Lecture Notes in Computer Science Sanskrit Computational Linguistics, (2008), p. 134–143, https://doi.org/10.1007/978-3-540-93885-9_11.
[9] V. K. Gupta, N. Tapaswi, and S. Jain, *Knowledge representation of grammatical constructs of sanskrit language using rule based sanskrit language to english language machine translation*, 2013 International Conference on Advances in Technology and Engineering (ICATE), (2013), https://doi.org/10.1109/icadte.2013.6524744.
[10] A. Hurskainen and J. Tiedemann, *Rule-based machine translation from english to finnish*, Proceedings of the Second Conference on Machine Translation, (2017), https://doi.org/10.18653/v1/w17-4731.
[11] L. Jain and P. Agrawal, *Text independent root word identification in hindi language using natural language processing*, International Journal of Advanced Intelligence Paradigms, 7 (2015), p. 240, https://doi.org/10.1504/ijaip.2015.073705.
[12] L. Jain and P. Agrawal, *English to sanskrit transliteration: an effective approach to design natural language translation tool*, International Journal of Advanced Research in Computer Science (IJARCS, 8 (2017), pp. 1–10, https://doi.org/https://doi.org/10.26483/ijarcs.v8i1.2860.
[13] J. K. and J. R., *Sanskrit machine translation systems: A comparative analysis*, International Journal of Computer Applications, 136 (2016), p. 1–4, https://doi.org/10.5120/ijca2016908290.
[14] A. Kumar, Saurabh, and M. Raza, *Syntax and semantic analysis of devanagari hindi*, International Journal of Recent Scientific Research, 8 (2017).
[15] V. Macketanz, E. Avramidis, A. Burchardt, J. Helcl, and A. Srivastava, *Machine translation: Phrase-based, rule-based and neural approaches with linguistic evaluation*, Cybernetics and Information Technologies, 17 (2017), p. 28–43, https://doi.org/10.1515/cait-2017-0014.
[16] S. Mall and U. C. Jaiswal, *Developing a system for machine translation from hindi language to english language*, 2013 4th International Conference on Computer and Communication Technology (ICCCT), (2013), https://doi.org/10.1109/iccct.2013.6749607.
[17] V. Mishra and R. Mishra, *English to sanskrit machine translation system: a rule-based approach*, International Journal of Advanced Intelligence Paradigms, 4 (2012), p. 168, https://doi.org/10.1504/ijaip.2012.048144.
[18] V. Mishra and R. B. Mishra, *Study of example based english to sanskrit machine translation*, Polibits, 37 (2008), p. 43–54, https://doi.org/10.17562/pb-37-5.
[19] A. Nayyar, B. Mahapatra, D. N. Le, and G. Suseendran, *Virtual reality (vr) & augmented reality (ar) technologies for tourism and hospitality industry*, International Journal of Engineering & Technology, 7 (2018).
[20] D. Q. Nguyen, D. Q. Nguyen, D. D. Pham, and S. B. Pham, *A robust transformation-based learning approach using ripple down rules for part-of-speech tagging*, AI Communications, 29 (2016), p. 409–422, https://doi.org/10.3233/aic-150698.
[21] W. P. Pa, Y. K. Thu, A. Finch, and E. Sumita, *A study of statistical machine translation methods for under resourced languages*, Procedia Computer Science, 81 (2016), p. 250–257, https://doi.org/10.1016/j.procs.2016.04.057.
[22] M. Parkvall, *Världens 100 största språk 2007 (the world's 100 largest languages in 2007)*, 2007.
[23] S. Ramteke, K. Ramteke, and D. R., *Lexicon parser for syntactic and semantic analysis of devanagari sentence using hindi wordnet*, International Journal of Advanced Research in Computer and Communication Engineering, 3 (2014).
[24] S. Saini and V. Sahula, *A survey of machine translation techniques and systems for indian languages*, 2015 IEEE International Conference on Computational Intelligence and Communication Technology, (2015), https://doi.org/10.1109/cict.2015.123.
[25] S. S. Sehra and A. Nayyar, *Paper on algorithms used for text classification*, 2013.
[26] N. Sethi, P. Agrawal, V. Madaan, and S. K. Singh, *A novel approach to paraphrase hindi sentences using natural language processing*, Indian Journal of Science and Technology, 9 (2016), https://doi.org/10.17485/ijst/2016/v9i28/98374.
[27] R. Singh and A. Nayyar, *Ant colony optimization — computational swarm intelligence technique*, (2016).
[28] H. Somers, *Review article: Example-based machine translation*, Machine Translation, 14 (1999), pp. 113–157, https://doi.org/10.1023/A:1008109312730, https://doi.org/10.1023/A:1008109312730.
[29] S. Sreelekha, P. Bhattacharyya, and D. Malathi, *Statistical vs. rule-based machine translation: A comparative study on indian languages*, Advances in Intelligent Systems and Computing International Conference on Intelligent Computing and Applications, (2017), p. 663–675, https://doi.org/10.1007/978-981-10-5520-1_59.

*Edited by:* Anand Nayyar
*Received:* Jun 21, 2020
*Accepted:* Jul 27, 2020